



Multiple-relaxation-time finite-difference lattice Boltzmann model for the nonlinear convection-diffusion equation

Xinmeng Chen,¹ Zhenhua Chai ^{1,2}, Jinlong Shang,¹ and Baochang Shi^{1,2,*}

¹*School of Mathematics and Statistics, Huazhong University of Science and Technology, Wuhan 430074, China*

²*Hubei Key Laboratory of Engineering Modeling and Scientific Computing, Huazhong University of Science and Technology, Wuhan 430074, China*

 (Received 24 May 2021; revised 19 August 2021; accepted 14 September 2021; published 29 September 2021)

In this paper, a multiple-relaxation-time finite-difference lattice Boltzmann method (MRT-FDLBM) is developed for the nonlinear convection-diffusion equation (NCDE). Through designing the equilibrium distribution function and the source term properly, the NCDE can be recovered exactly from MRT-FDLBM. We also conduct the von Neumann stability analysis on the present MRT-FDLBM and its special case, i.e., single-relaxation-time finite-difference lattice Boltzmann method (SRT-FDLBM). Then, a simplified version of MRT-FDLBM (SMRT-FDLBM) is also proposed, which can save about 15% computational cost. In addition, a series of real and complex-value NCDEs, including the isotropic convection-diffusion equation, Burgers-Fisher equation, sine-Gordon equation, heat-conduction equation, and Schrödinger equation, are used to test the performance of MRT-FDLBM. The results show that both MRT-FDLBM and SMRT-FDLBM have second-order convergence rates in space and time. Finally, the stability and accuracy of five different models are compared, including the MRT-FDLBM, SMRT-FDLBM, SRT-FDLBM, the previous finite-difference lattice Boltzmann method [H. Wang, B. Shi *et al.*, *Appl. Math. Comput.* **309**, 334 (2017)], and the lattice Boltzmann method (LBM). The stability tests show that the sequence of stability from high to low is as follows: MRT-FDLBM, SMRT-FDLBM, SRT-FDLBM, the previous finite-difference lattice Boltzmann method, and LBM. In most of the precision test results, it is found that the order from high to low of precision is MRT-FDLBM, SMRT-FDLBM, SRT-FDLBM, and the previous finite-difference lattice Boltzmann method.

DOI: [10.1103/PhysRevE.104.035308](https://doi.org/10.1103/PhysRevE.104.035308)

I. INTRODUCTION

The nonlinear convection-diffusion equation (NCDE) is widely applied to describe the physical phenomena, such as heat and mass transfer [1]. However, due to the existence of the nonlinearity, it is difficult to get the analytical solution of NCDE. For this reason, some efficient numerical methods have been developed to solve NCDEs, including the finite-volume method [2], finite-element method [3], finite-difference method [4], and so on.

Since the early 1990s, the lattice Boltzmann method (LBM) and discrete Boltzmann modeling method (DBM) have been remarkably successful in the study of complex fluid problems, such as fluid flows in porous media [5,6], multiphase flow [7–10], turbulence [11,12], combustion [13,14] and hydrodynamic instability [15,16]. At the same time, many different LBMs have also been developed to solve the convection-diffusion equations. Ponce Dawson *et al.* [17] first proposed the LBM with a linear equilibrium distribution function for reaction-diffusion systems, while for CDE, the LBM cannot give correct macroscopic equation. To recover the general NCDE exactly, Shi *et al.* [18] introduced an auxiliary moment in the LBM, but this model has a special requirement on the convection term. Chopard [19] also developed

a new LBM for NCDE, where a temporal derivative was introduced to recover NCDE exactly. Yoshida and Nagaoka [20] proposed a multiple-relaxation-time (MRT) LBM, and the NCDE can be recovered correctly by MRT-LBM under the diffusive scaling. Chai *et al.* [21] further developed a general MRT LBM for nonlinear anisotropic convection-diffusion equation (NACDE) without any assumptions on diffusion and convection terms. Besides, Ginzburg also proposed a two-relaxation-time LBM (TRT-LBM) and the MRT-LBM for NACDE and presented some analysis on them [22–25]. Zhao *et al.* proposed a block triple-relaxation-time lattice Boltzmann model for general NACDEs [26]. Recently, Chai and Shi [27] developed a unified framework of MRT-LBM for Navier-Stokes equations and NCDE by introducing a more general collision term and an auxiliary source distribution function. These LBMs could solve NCDE well, but they only have first-order convergence rate in time, and it is also difficult to implement them on the nonuniform grid.

To overcome above problems, some approaches have been proposed, including LBM combined with interpolation approach [28,29] and rectangular LBM [30–32], but these two kinds of the LBM are still first-order accuracy in time. In addition, as an efficient alternative, the finite-difference lattice Boltzmann method (FDLBM) not only can be implemented directly in nonuniform grids but also can be used improve the accuracy in time through introducing some special schemes for time derivative. In the year of 1995, the finite-difference

*Corresponding author: shibc@hust.edu.cn

lattice Boltzmann equation was first proposed by Reider and Sterling [33]. Then Mei and Shyy [34] developed a FDLBM with the body-fitted grid. Junk and Klar [35] derived a relaxation system for an equivalent set of velocity moments, and a simple upwind scheme is used to give rise to the spatial discretization of the moment system. Based on these previous works, Guo *et al.* [36] proposed an improved FDLBM, and used a mixed difference scheme to discretize the gradient operator. In addition, many high-order accuracy FDLBMs are also developed [37–39], and have been used to simulate complex flow problems, including three-dimensional incompressible flows [40], two-phase liquid-vapor flows [41], natural convection in some special geometries [42,43], and blood flow [44]. Recently, Chen *et al.* [45] developed a second-order accurate FDLBM for incompressible flows; this FDLBM not only keeps the simplicity of LBM but also can be used to improve the computational efficiency. However, it should be noted that all above works focused on the fluid flows; only Wang *et al.* [46] proposed the FDLBM with a mixed difference scheme for NCDE, and this FDLBM can recover the NCDE correctly. However, there are three limitations in the FDLBM of Wang *et al.* [46]. The first is that the method has only first-order accuracy in time. Second, the value of the Courant-Friedrichs-Lewy (CFL) condition number is usually about 0.1 or 0.2, which makes the computational cost more expensive. And, third, the collision terms in all above FDLBMs are approximated by the Bhatnagar-Gross-Krook (BGK) model, which may bring the numerical instability problem. Inspired by the previous works [27,45],

we developed a MRT-FDLBM for the NCDE where the MRT model combined with a second-order accurate FDLBM are considered. Furthermore, the MRT-FDLBM can recover NCDE exactly by designing the equilibrium distribution function and the source term appropriately.

The rest of the paper is organized as follows. A new MRT-FDLBM is proposed for NCDE in Sec. 2. In Sec. 3, we show that the NCDE can be recovered from the MRT-FDLBM through the Champman-Enskog (CE) analysis. Then the stability of MRT-FDLBM is investigated in Sec. 4. In addition, the SMRT-FDLBM is presented in Sec. 5. In Sec. 6, we extend the MRT-FDLBM to the complex-valued NCDE. In Sec. 7, a series of numerical examples are studied to test the accuracy and convergence rate of the present MRT-FDLBM. Finally, we make a brief summary in Sec. 8.

II. MULTIPLE-RELAXATION-TIME FINITE-DIFFERENCE LATTICE BOLTZMANN MODEL

A. Nonlinear convection-diffusion equation

The d -dimensional NCDE with a source term can be expressed as

$$\partial_t \phi + \nabla \cdot \mathbf{B} = \nabla \cdot [\mathbf{K} \cdot (\nabla \cdot \mathbf{D})] + R, \quad (1)$$

where ϕ is scalar variable related to time t and position \mathbf{x} ; \mathbf{B} is the convection term; \mathbf{D} is the diffusion term; R is the source term, and usually they are the function of ϕ , \mathbf{x} , and t ; ∇ is the gradient operator; and $\mathbf{K} = \mathbf{K}(\phi, \mathbf{x}, t)$ is the diffusion tensor.

B. Multiple-relaxation-time finite-difference lattice Boltzmann method

Following the idea of the previous work [45], we consider a second-order accurate MRT-FDLBM. For generality, the general evolution equation with DdQq model (q discrete velocities in d -dimensional space) can be written as

$$\begin{aligned} f_j(\mathbf{x}, t + \Delta t) - f_j(\mathbf{x}, t) + \tilde{B}_0 \Delta t \mathbf{c}_j \cdot \nabla f_j(\mathbf{x}, t) + \tilde{B}_1 \Delta t \mathbf{c}_j \cdot \nabla f_j(\mathbf{x}, t + A \Delta t) \\ + \tilde{B}_2 \Delta t \mathbf{c}_j \cdot \nabla f_j(\mathbf{x}, t + \Delta t) = \tilde{B}_0 \Delta t (-\tilde{\Lambda}_{jk} f_k^{\text{ne}})(\mathbf{x}, t) + \tilde{B}_1 \Delta t (-\tilde{\Lambda}_{jk} f_k^{\text{ne}})(\mathbf{x}, t + A \Delta t) \\ + \tilde{B}_2 \Delta t (-\tilde{\Lambda}_{jk} f_k^{\text{ne}})(\mathbf{x}, t + \Delta t) + \Delta t \left[F_j(\mathbf{x}, t) + G_j(\mathbf{x}, t) + \frac{\Delta t}{2} \theta \partial_t F_j(\mathbf{x}, t) + \frac{\Delta t}{2} \bar{\theta} \partial_t G_j(\mathbf{x}, t) \right]. \end{aligned} \quad (2)$$

where θ and $\bar{\theta}$ are two tunable parameters to be specified in the next section. $\tilde{\Lambda} = (\tilde{\Lambda}_{jk})$ is a $q \times q$ invertible collision matrix. The other parameters should satisfy

$$\tilde{B}_0 + \tilde{B}_1 + \tilde{B}_2 = 1, \quad A \tilde{B}_1 + \tilde{B}_2 = \frac{1}{2}, \quad (3a)$$

$$\tilde{B}_0 + \tilde{B}_1 + \tilde{B}_2 = 1, \quad A \tilde{B}_1 + \tilde{B}_2 = \frac{1}{2}. \quad (3b)$$

$f_j(\mathbf{x}, t)$ is the distribution function at time t and location \mathbf{x} . Δt is the time step, and $f_k^{\text{ne}}(\mathbf{x}, t) = f_k(\mathbf{x}, t) - f_k^{\text{eq}}(\mathbf{x}, t)$ with $f_k^{\text{eq}}(\mathbf{x}, t)$ being the equilibrium distribution function,

$$f_k^{\text{eq}}(\mathbf{x}, t) = \omega_j \left[\phi + \frac{\mathbf{c}_j \cdot \mathbf{B}}{c_s^2} + \frac{(\beta c_s^2 \mathbf{D} + \mathbf{C} - c_s^2 \phi \mathbf{I}) : (\mathbf{c}_j \mathbf{c}_j - c_s^2 \mathbf{I})}{2c_s^4} \right], \quad (4)$$

where \mathbf{I} is the unit matrix, \mathbf{c}_j is the discrete velocity, ω_j is the weight coefficient, and the positive parameter β is related to the diffusion tensor \mathbf{K} .

The discrete source terms F_j and G_j are defined as

$$F_j(\mathbf{x}, t) = \omega_j \left(R + \frac{\mathbf{c}_j \cdot \mathbf{M}_{1,F}}{c_s^2} \right), \quad (5a)$$

$$G_j(\mathbf{x}, t) = \omega_j \frac{\mathbf{c}_j \cdot \mathbf{M}_{1,G}}{c_s^2}, \quad (5b)$$

To derive the NCDE correctly, the following relations should be satisfied:

$$\sum_j f_j = \sum_j f_j^{\text{eq}} = \phi, \quad \sum_j c_j f_j^{\text{eq}} = \mathbf{B}, \quad \sum_j c_j c_j f_j^{\text{eq}} = \beta c_s^2 \mathbf{D} + \mathbf{C}, \quad (6a)$$

$$\sum_j F_j = R, \quad \sum_j c_j F_j = \mathbf{M}_{1,F}, \quad \sum_j G_j = 0, \quad \sum_j c_j G_j = \mathbf{M}_{1,G}, \quad (6b)$$

where $\mathbf{M}_{1,F}$ and $\mathbf{M}_{1,G}$ will be determined in Sec. 3. In addition, the second evolution equation used to evaluate distribution function $f_j(\mathbf{x}, t + \frac{1}{2}\Delta t)$ can be written as

$$f_j(\mathbf{x}, t + h) - f_j(\mathbf{x} - \mathbf{c}_j h, t) = \frac{h}{2} \left[-\tilde{\Lambda}_{jk} f_k^{\text{ne}}(\mathbf{x}, t + h) - \tilde{\Lambda}_{jk} f_k^{\text{ne}}(\mathbf{x} - \mathbf{c}_j h, t) \right] + h \left[F_j(\mathbf{x} - \mathbf{c}_j h, t) + \tilde{G}_j(\mathbf{x} - \mathbf{c}_j h, t) + \frac{h}{2} \bar{D}_j F_j(\mathbf{x} - \mathbf{c}_j h, t) + \frac{h}{2} \bar{D}_j \tilde{G}_j(\mathbf{x} - \mathbf{c}_j h, t) \right], \quad (7)$$

where $h = \Delta t/2$, $\bar{D}_j = \theta_1 \partial_t + \gamma_1 \mathbf{c}_j \cdot \nabla$, $\tilde{D}_j = \theta_2 \partial_t + \gamma_2 \mathbf{c}_j \cdot \nabla$. The parameters θ_1 , θ_2 , γ_1 , and γ_2 will be determined in Sec. 3. The discrete source terms \tilde{G}_j is given by

$$\tilde{G}_j(\mathbf{x}, t) = \omega_j \frac{\mathbf{c}_j \cdot \mathbf{M}_{1,G}}{c_s^2}, \quad (8)$$

where $\mathbf{M}_{1,G}$ will also be given in Sec. 3.

C. Computational procedure of the second-order multiple-relaxation-time finite-difference lattice Boltzmann method

In the computational process, the parameters are taken the same values ($A = \frac{1}{2}$, $\tilde{B}_0 = 0$, $\tilde{B}_1 = 1$, $\tilde{B}_2 = 0$, $\bar{B}_0 = \frac{1}{2}$, $\bar{B}_1 = 0$, $\bar{B}_2 = \frac{1}{2}$) as those in Ref. [45], then the first evolution Eq. (2) can be rewritten as

$$f_j(\mathbf{x}, t + \Delta t) - f_j(\mathbf{x}, t) + \Delta t \mathbf{c}_j \cdot \nabla f_j \left(\mathbf{x}, t + \frac{1}{2} \Delta t \right) = \frac{\Delta t}{2} \left[-\tilde{\Lambda}_{jk} f_k^{\text{ne}}(\mathbf{x}, t + \Delta t) - \tilde{\Lambda}_{jk} f_k^{\text{ne}}(\mathbf{x}, t) \right] + \Delta t \left[F_j(\mathbf{x}, t) + G_j(\mathbf{x}, t) + \frac{\Delta t}{2} \theta \partial_t F_j(\mathbf{x}, t) + \frac{\Delta t}{2} \bar{\theta} \partial_t G_j(\mathbf{x}, t) \right]. \quad (9)$$

After some manipulations, Eq. (9) can be implemented explicitly with the following form:

$$\hat{f}_j(\mathbf{x}, t + \Delta t) + \Delta t \mathbf{c}_j \cdot \nabla f_j \left(\mathbf{x}, t + \frac{1}{2} \Delta t \right) = \hat{f}_j^+(\mathbf{x}, t) + \Delta t \left[F_j(\mathbf{x}, t) + \frac{\Delta t}{2} \theta \partial_t F_j(\mathbf{x}, t) \right] + \Delta t \left[G_j(\mathbf{x}, t) + \frac{\Delta t}{2} \bar{\theta} \partial_t G_j(\mathbf{x}, t) \right], \quad (10)$$

where

$$\hat{f}_j = f_j - \frac{1}{2} \Delta t \left[-\tilde{\Lambda}_{jk} (f_k - f_k^{\text{eq}}) \right], \quad (11a)$$

$$\hat{f}_j^+ = f_j + \frac{1}{2} \Delta t \left[-\tilde{\Lambda}_{jk} (f_k - f_k^{\text{eq}}) \right]. \quad (11b)$$

Equations (11a) and (11b) can also be expressed in the matrix form,

$$\hat{\mathbf{f}} = \mathbf{f} - \frac{1}{2} \Delta t \left[-\tilde{\Lambda} (\mathbf{f} - \mathbf{f}^{\text{eq}}) \right] = \left(\mathbf{I} + \frac{\Delta t}{2} \tilde{\Lambda} \right) \mathbf{f} - \frac{\Delta t}{2} \tilde{\Lambda} \mathbf{f}^{\text{eq}}, \quad (12a)$$

$$\hat{\mathbf{f}}^+ = \left(\mathbf{I} - \frac{\Delta t}{2} \tilde{\Lambda} \right) \mathbf{f} + \frac{\Delta t}{2} \tilde{\Lambda} \mathbf{f}^{\text{eq}}, \quad (12b)$$

or, equivalently,

$$\mathbf{f} = \left(\mathbf{I} + \frac{\Delta t}{2} \tilde{\Lambda} \right)^{-1} \left(\hat{\mathbf{f}} + \frac{\Delta t}{2} \tilde{\Lambda} \mathbf{f}^{\text{eq}} \right), \quad (13a)$$

$$\hat{\mathbf{f}}^+ = \left(\mathbf{I} - \frac{\Delta t}{2} \tilde{\Lambda} \right) \left(\mathbf{I} + \frac{\Delta t}{2} \tilde{\Lambda} \right)^{-1} \left(\hat{\mathbf{f}} + \frac{\Delta t}{2} \tilde{\Lambda} \mathbf{f}^{\text{eq}} \right) + \frac{\Delta t}{2} \tilde{\Lambda} \mathbf{f}^{\text{eq}}. \quad (13b)$$

Let

$$\mathbf{I} - \mathbf{\Lambda} = \left(\mathbf{I} - \frac{\Delta t}{2} \tilde{\mathbf{\Lambda}} \right) \left(\mathbf{I} + \frac{\Delta t}{2} \tilde{\mathbf{\Lambda}} \right)^{-1}. \quad (14)$$

Equation (13b) can be rearranged as

$$\hat{\mathbf{f}}^+ = (\mathbf{I} - \mathbf{\Lambda})\hat{\mathbf{f}} + \mathbf{\Lambda}\mathbf{f}^{\text{eq}}. \quad (15)$$

Similarly, the second evolution equation (7) can also be simplified into an explicit form,

$$\bar{f}_j(\mathbf{x}, t + h) = \bar{f}_j^+(\mathbf{x} - \mathbf{c}_j h, t) + h \left[F_j + G_j + \frac{h}{2} \bar{D}_j F_j + \frac{h}{2} \bar{\bar{D}}_j G_j \right] (\mathbf{x} - \mathbf{c}_j h, t), \quad (16)$$

where

$$\bar{f}_j = f_j - \frac{h}{2} \left[-\tilde{\Lambda}_{jk} (f_k - f_k^{\text{eq}}) \right], \quad (17a)$$

$$\bar{f}_j^+ = f_j + \frac{h}{2} \left[-\tilde{\Lambda}_{jk} (f_k - f_k^{\text{eq}}) \right]. \quad (17b)$$

Similarly to above discussion, we can also write Eqs. (17a) and (17b) in the matrix form,

$$\bar{\mathbf{f}} = \mathbf{f} - \frac{h}{2} \left[-\tilde{\mathbf{\Lambda}}(\mathbf{f} - \mathbf{f}^{\text{eq}}) \right] = \left(\mathbf{I} + \frac{h}{2} \tilde{\mathbf{\Lambda}} \right) \mathbf{f} - \frac{h}{2} \tilde{\mathbf{\Lambda}} \mathbf{f}^{\text{eq}}, \quad (18a)$$

$$\bar{\mathbf{f}}^+ = \left(\mathbf{I} - \frac{h}{2} \tilde{\mathbf{\Lambda}} \right) \mathbf{f} + \frac{h}{2} \tilde{\mathbf{\Lambda}} \mathbf{f}^{\text{eq}}, \quad (18b)$$

or, equivalently,

$$\mathbf{f} = \left(\mathbf{I} + \frac{h}{2} \tilde{\mathbf{\Lambda}} \right)^{-1} \left(\bar{\mathbf{f}} + \frac{h}{2} \tilde{\mathbf{\Lambda}} \mathbf{f}^{\text{eq}} \right), \quad (19)$$

$$\bar{\mathbf{f}}^+ = \left(\mathbf{I} - \frac{3\tilde{\mathbf{\Lambda}}}{4} \right) \hat{\mathbf{f}} + \frac{3\tilde{\mathbf{\Lambda}}}{4} \mathbf{f}^{\text{eq}}. \quad (20)$$

Applying Taylor expansion to Eq. (16) and ignoring the term $O(h^2)$, we have

$$\bar{f}_j(\mathbf{x}, t + h) = \bar{f}_j^+(\mathbf{x}, t) - h \mathbf{c}_j \cdot \nabla \bar{f}_j^+(\mathbf{x}, t) + h \left[F_j(\mathbf{x}, t) + \bar{G}_j(\mathbf{x}, t) + \frac{h}{2} \bar{D}_j F_j(\mathbf{x}, t) + \frac{h}{2} \bar{\bar{D}}_j \bar{G}_j(\mathbf{x}, t) \right]. \quad (21)$$

The gradient terms ∇f_j in Eq. (10) and $\nabla \bar{f}_j^+$ in Eq. (21) can be discretized by a mixed difference scheme [36],

$$\nabla \Pi_j^* = \frac{\partial \Pi_j^*}{\partial \chi_\alpha} \Big|_m = \eta \frac{\partial \Pi_j^*}{\partial \chi_\alpha} \Big|_c + (1 - \eta) \frac{\partial \Pi_j^*}{\partial \chi_\alpha} \Big|_u, \quad (22)$$

where Π_j^* represents f_j or \bar{f}_j^+ , and the parameter $\eta \in [0, 1]$. The terms $\frac{\partial \Pi_j^*}{\partial \chi_\alpha} \Big|_u$ and $\frac{\partial \Pi_j^*}{\partial \chi_\alpha} \Big|_c$ represent second up-wind difference and central-difference schemes and are given by

$$\frac{\partial \Pi_j^*}{\partial \chi_\alpha} \Big|_c = \frac{\Pi_j^*(\chi_\alpha + \Delta \chi_\alpha, t) - \Pi_j^*(\chi_\alpha - \Delta \chi_\alpha, t)}{2\Delta \chi_\alpha}, \quad (23a)$$

$$\frac{\partial \Pi_j^*}{\partial \chi_\alpha} \Big|_u = \begin{cases} \frac{3\Pi_j^*(\chi_\alpha, t) - 4\Pi_j^*(\chi_\alpha - \Delta \chi_\alpha, t) + \Pi_j^*(\chi_\alpha - 2\Delta \chi_\alpha, t)}{2\Delta \chi_\alpha}, & \text{if } c_{i\alpha} \geq 0, \\ \frac{-3\Pi_j^*(\chi_\alpha, t) + 4\Pi_j^*(\chi_\alpha + \Delta \chi_\alpha, t) - \Pi_j^*(\chi_\alpha + 2\Delta \chi_\alpha, t)}{2\Delta \chi_\alpha}, & \text{if } c_{i\alpha} < 0. \end{cases} \quad (23b)$$

The evolution process of the MRT-FDLBM is shown in Fig. 1 and can be listed as follows:

Step (1): Calculate $\hat{\mathbf{f}}^+(\mathbf{x}, t)$ by Eq. (15).

Step (2): Evaluate spatial gradient term $\mathbf{c} \cdot \nabla \mathbf{f}(\mathbf{x}, t + h)$,

$$\hat{\mathbf{f}}(\mathbf{x}, t) \xrightarrow{(20)} \bar{\mathbf{f}}^+(\mathbf{x}, t) \xrightarrow{(21)} \bar{\mathbf{f}}(\mathbf{x}, t + h) \xrightarrow{(19)} \mathbf{f}(\mathbf{x}, t + h) \xrightarrow{(22)} \nabla \mathbf{f}(\mathbf{x}, t + h),$$

Step (3): Compute $\hat{\mathbf{f}}(\mathbf{x}, t + \Delta t)$ by Eq. (10).

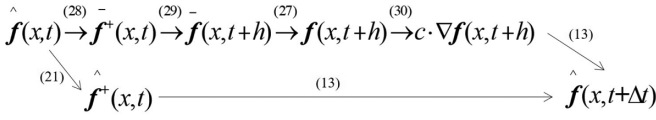


FIG. 1. The implementation process of the MRT-FDLBM.

III. THE CHAPMAN-ENSKOG ANALYSIS OF THE PRESENT MRT-FDLBM

In this part, we will perform a detailed CE analysis to recover the NCDE from the present MRT-FDLBM. In the standard LBM, there is only an evolution equation for macroscopic equation, while in the present MRT-FDLBM, there are two evolution Eqs. (2) and (7). Although the principle of CE

analysis of two evolution equations are similar, the moment conditions of the source terms are different from each other.

We first conducted the Chapman-Enskog analysis to the main evolution Eq. (2) to recover the NCDE. The distribution function f_j , the source terms, the time and space derivatives can be expanded as

$$\begin{aligned} f_j &= f_j^{(0)} + \varepsilon f_j^{(1)} + \varepsilon^2 f_j^{(2)}, \quad \partial_t = \varepsilon \partial_{t_1} + \varepsilon^2 \partial_{t_2}, \quad \nabla = \varepsilon \nabla_1, \\ \tilde{G}_j &= \varepsilon \tilde{G}_j^{(1)} + \varepsilon^2 \tilde{G}_j^{(2)}, \quad F_j = \varepsilon F_j^{(1)} + \varepsilon^2 F_j^{(2)}. \end{aligned} \quad (24)$$

Applying the Taylor expansion to Eq. (2) gives rise to the following equation:

$$\begin{aligned} \Delta t \partial_t f_j + \frac{\Delta t^2}{2} \partial_{t_1}^2 f_j + \tilde{B}_0 \Delta t \mathbf{c}_j \cdot \nabla f_j + \tilde{B}_1 \Delta t \mathbf{c}_j \cdot \nabla f_j + \tilde{B}_1 \Delta t \mathbf{c}_j \cdot \nabla A \Delta t \partial_t f_j + \tilde{B}_2 \Delta t \mathbf{c}_j \cdot \nabla f_j \\ + \tilde{B}_2 \Delta t \mathbf{c}_j \cdot \nabla \Delta t \partial_t f_j = \Delta t (-\tilde{\Lambda}_{jk} f_k^{\text{neq}}) + \frac{\Delta t^2}{2} \partial_{t_1} (-\tilde{\Lambda}_{jk} f_k^{\text{neq}}) + \Delta t F_j + \frac{\Delta t^2}{2} \partial_{t_1} F_j, \end{aligned} \quad (25)$$

which can be simplified by

$$\partial_t f_j + \frac{\Delta t}{2} \partial_{t_1}^2 f_j + \mathbf{c}_j \cdot \nabla f_j + \mathbf{c}_j \cdot \nabla \frac{\Delta t}{2} \partial_t f_j = [-\tilde{\Lambda}_{jk} (f_k - f_k^{\text{eq}})] + \frac{\Delta t}{2} \partial_{t_1} [-\tilde{\Lambda}_{jk} (f_k - f_k^{\text{eq}})] + F_j + G_j + \frac{\Delta t}{2} (\theta \partial_{t_1} F_j + \bar{\theta} \partial_{t_1} G_j). \quad (26)$$

Substituting Eq. (24) into Eq. (26) yields

$$O(\varepsilon^0) : -\tilde{\Lambda}_{jk} [f_k^{(0)} - f_k^{\text{eq}}] = 0 \Leftrightarrow f_j^{(0)} = f_j^{\text{eq}}, \quad (27a)$$

$$O(\varepsilon^1) : \partial_{t_1} f_j^{(0)} + \mathbf{c}_j \cdot \nabla_1 f_j^{(0)} = -\tilde{\Lambda}_{jk} f_k^{(1)} + G_j^{(1)} + F_j^{(1)}, \quad (27b)$$

$$\begin{aligned} O(\varepsilon^2) : \partial_{t_2} f_j^{(0)} + \partial_{t_1} f_j^{(1)} + \frac{\Delta t}{2} \partial_{t_1}^2 f_j^{(0)} + \mathbf{c}_j \cdot \nabla_1 f_j^{(1)} + \mathbf{c}_j \cdot \nabla_1 \frac{\Delta t}{2} \partial_{t_1} f_j^{(0)} \\ = -\tilde{\Lambda}_{jk} f_k^{(2)} + \frac{\Delta t}{2} \partial_{t_1} [-\tilde{\Lambda}_{jk} f_k^{(1)}] + G_j^{(2)} + F_j^{(2)} + \frac{\Delta t}{2} [\theta \partial_{t_1} F_j^{(1)} + \bar{\theta} \partial_{t_1} G_j^{(1)}]. \end{aligned} \quad (27c)$$

For the collision matrix $\tilde{\Lambda}$, the following requirements should be satisfied [27]:

$$\sum_j \mathbf{e}_j \tilde{\Lambda}_{jk} = s_0 \mathbf{e}_k, \quad \sum_j \mathbf{c}_j \tilde{\Lambda}_{jk} = \tilde{\mathbf{S}} \mathbf{c}_k, \quad \forall k = 1, 2, \dots, q, \quad (28)$$

where $\mathbf{e} = (1, 1, \dots, 1) \in \mathbb{R}^q$, $\tilde{\mathbf{S}}$ is an invertible $d \times d$ relaxation matrix corresponding to the diffusion tensor \mathbf{K} . Based on the Eqs. (6a), (6b), and (28), we can get the following equations through summing Eqs. (27b) and (27c) over j :

$$\partial_{t_1} \phi + \nabla_1 \cdot \mathbf{B} = R^{(1)}, \quad (29a)$$

$$\partial_{t_2} \phi + \frac{\Delta t}{2} \partial_{t_1}^2 \phi + \nabla_1 \cdot \sum_j \mathbf{c}_j f_j^{(1)} + \frac{\Delta t}{2} \nabla_1 \cdot \partial_{t_1} \mathbf{B} = R^{(2)} + \frac{\Delta t}{2} \theta \partial_{t_1} R^{(1)}. \quad (29b)$$

According to Eq. (29a), one can obtain

$$\partial_{t_1}^2 \phi = \partial_{t_1} R^{(1)} - \partial_{t_1} \nabla_1 \cdot \mathbf{B}. \quad (30)$$

If the derivatives of time and the space are interchangeable and $\theta = 1$, Eq. (29b) can be rewritten as

$$\partial_{t_2} \phi = R^{(2)} - \nabla_1 \cdot \sum_j \mathbf{c}_j f_j^{(1)}. \quad (31)$$

With the aid of Eq. (28), we have the following equation through multiplying the Eq. (27b) by \mathbf{c}_j and summing it over j :

$$\partial_{t_1} \mathbf{B} + \nabla_1 \cdot (\beta c_s^2 \mathbf{D} + \mathbf{C}) = -\tilde{\mathbf{S}} \sum_j \mathbf{c}_j f_j^{(1)} + \mathbf{M}_{1,G}^{(1)} + \mathbf{M}_{1,F}^{(1)}, \quad (32)$$

where $\mathbf{M}_{1,F}^{(1)} = \sum_j \mathbf{c}_j F_j^{(1)}$, $\mathbf{M}_{1,G}^{(1)} = \sum_j \mathbf{c}_j G_j^{(1)}$, and the diagonal matrix $\tilde{\mathbf{S}}$ is given in Appendix. From Eq. (32), one can obtain

$$\sum_j \mathbf{c}_j f_j^{(1)} = -\tilde{\mathbf{S}}^{-1} [\partial_{t_1} \mathbf{B} + \nabla_1 \cdot (\beta c_s^2 \mathbf{D} + \mathbf{C}) - \mathbf{M}_{1,G}^{(1)} - \mathbf{M}_{1,F}^{(1)}], \quad (33)$$

and

$$\partial_{t_2}\phi = R^{(2)} + \nabla_1 \cdot \beta c_s^2 \tilde{\mathcal{S}}^{-1} \nabla_1 \cdot \mathbf{D} + \nabla_1 \cdot \tilde{\mathcal{S}}^{-1} [\partial_{t_1} \mathbf{B} + \nabla_1 \cdot \mathbf{C} - \mathbf{M}_{1,F}^{(1)} - \mathbf{M}_{1,G}^{(1)}]. \quad (34)$$

Let

$$\mathbf{RH} = \tilde{\mathcal{S}}^{-1} (\partial_t \mathbf{B} + \nabla \cdot \mathbf{C} - \mathbf{M}_{1,F} - \mathbf{M}_{1,G}), \quad (35)$$

if $\mathbf{RH} = 0$ and $\mathbf{K} = \beta c_s^2 \tilde{\mathcal{S}}^{-1}$, Eq. (34) would reduce to

$$\partial_{t_2}\phi = R^{(2)} + \nabla_1 \cdot \mathbf{K} \cdot \nabla_1 \cdot \mathbf{D}. \quad (36)$$

Through combining the results at ε and ε^2 scales, i.e., Eqs. (29a) and (36), we can recover the NCDE (2) correctly.

We now present some discussion on two special cases under the condition of $\mathbf{RH} = 0$.

Scheme A: $\mathbf{B} = \mathbf{B}(\phi)$, $\mathbf{M}_{1,F} = 0$. Under these conditions, we have

$$\mathbf{RH} = \tilde{\mathcal{S}}^{-1} (\partial_t \mathbf{B} + \nabla \cdot \mathbf{C} - \mathbf{M}_{1,G}) = 0, \quad (37)$$

which leads to following result:

$$\mathbf{M}_{1,G} = \partial_t \mathbf{B} + \nabla \cdot \mathbf{C}, \quad (38)$$

where \mathbf{C} is set as

$$\mathbf{C}_{\alpha\beta} = \int \mathbf{B}'_{\alpha} \mathbf{B}'_{\beta} d\phi, \quad \text{or} \quad \frac{d\mathbf{C}_{\alpha\beta}}{d\phi} = \frac{d\mathbf{B}_{\alpha}}{d\phi} \frac{d\mathbf{B}_{\beta}}{d\phi}. \quad (39)$$

According to Eq. (29a), we have

$$\partial_{t_1} \mathbf{B} + \nabla_1 \cdot \mathbf{C} = \mathbf{B}' (\partial_{t_1} \phi + \nabla_1 \cdot \mathbf{B}) = \mathbf{B}' R^{(1)}, \quad (40)$$

and then $\mathbf{M}_{1,G}$ can be determined by

$$\mathbf{M}_{1,G} = \mathbf{B}' R. \quad (41)$$

Scheme B: $\mathbf{C} = 0$, $\mathbf{M}_{1,F} = 0$. Under these conditions, we can obtain

$$\mathbf{RH} = \tilde{\mathcal{S}}^{-1} (\partial_t \mathbf{B} - \mathbf{M}_{1,G}) = 0, \quad (42)$$

and then

$$\mathbf{M}_{1,G} = \partial_t \mathbf{B}. \quad (43)$$

Next, we will recover the NCDE from the second evolution Eq. (7) with the CE analysis. With the Taylor expansion, Eq. (7) can be expressed as

$$D_j f_j + \frac{\hbar}{2} D_j^2 f_j = -\tilde{\Lambda}_{jk} f_k^{\text{ne}} + \frac{\hbar}{2} (-\tilde{\Lambda}_{jk} D_k f_k^{\text{ne}}) + \tilde{G}_j + F_j + \frac{\hbar}{2} (\tilde{D}_j F_j + \tilde{D}_j \tilde{G}_j). \quad (44)$$

Then we can expand Eq. (44) at different orders of ε ,

$$O(\varepsilon^0) : -\tilde{\Lambda}_{jk} (f_k^{(0)} - f_k^{\text{eq}}) = 0, \Leftrightarrow f_j^{(0)} = f_j^{\text{eq}}, \quad (45a)$$

$$O(\varepsilon^1) : D_{1j} f_j^{(0)} = -\tilde{\Lambda}_{jk} f_k^{(1)} + \tilde{G}_j^{(1)} + F_j^{(1)}, \quad (45b)$$

$$O(\varepsilon^2) : \partial_{t_2} f_j^{(0)} + D_{1j} f_j^{(1)} + \frac{\hbar}{2} D_{1j}^2 f_j^{(0)} = -\tilde{\Lambda}_{jk} f_k^{(2)} + \frac{\hbar}{2} [-\tilde{\Lambda}_{jk} D_{1k} f_k^{(1)}] + \tilde{G}_j^{(2)} + F_j^{(2)} \\ + \frac{\hbar}{2} [\tilde{D}_{1j} F_j^{(1)} + \tilde{D}_{1j} \tilde{G}_j^{(1)}]. \quad (45c)$$

With the help of Eq. (45b), Eq. (45c) can be rewritten as

$$\partial_{t_2} f_j^{(0)} + D_{1j} f_j^{(1)} = -\tilde{\Lambda}_{jk} f_k^{(2)} + \tilde{G}_j^{(2)} + F_j^{(2)} + \frac{\hbar}{2} [(\theta_1 - 1) \partial_{t_1} F_j^{(1)} + \nabla_1 \cdot \mathbf{c}_j (\gamma_1 - 1) F_j^{(1)} \\ + (\theta_2 - 1) \partial_{t_1} \tilde{G}_j^{(1)} + \nabla_1 \cdot \mathbf{c}_j (\gamma_2 - 1) \tilde{G}_j^{(1)}]. \quad (46)$$

Summing Eqs. (46) and (45b) over j , one can obtain

$$\partial_{t_1} \phi + \nabla_1 \cdot \mathbf{B} = R^{(1)}, \quad (47a)$$

$$\partial_{t_2} \phi + \nabla_1 \cdot \sum \mathbf{c}_j f_j^{(1)} = R^{(2)} + \frac{\hbar}{2} \{ (\theta_1 - 1) \partial_{t_1} R^{(1)} + \nabla_1 \cdot [(\gamma_1 - 1) \mathbf{M}_{1,F}^{(1)} + (\gamma_2 - 1) \mathbf{M}_{1,G}^{(1)}] \}. \quad (47b)$$

Multiplying the Eq. (45b) by \mathbf{c}_j and summing it over j , we can derive

$$\partial_t \mathbf{B} + \nabla_1 \cdot (\beta c_s^2 \mathbf{D} + \mathbf{C}) = -\tilde{\mathbf{S}} \sum \mathbf{c}_j f_j^{(1)} + \mathbf{M}_{1,\tilde{G}}^{(1)} + \mathbf{M}_{1,F}^{(1)}. \quad (48)$$

or

$$\sum \mathbf{c}_j f_j^{(1)} = -\tilde{\mathbf{S}}^{-1} [\partial_t \mathbf{B} + \nabla_1 \cdot (\beta c_s^2 \mathbf{D} + \mathbf{C}) - \mathbf{M}_{1,\tilde{G}}^{(1)} - \mathbf{M}_{1,F}^{(1)}], \quad (49)$$

where Eq. (28) has been used. Substituting above Eq. into Eq. (47b) and taking $\theta_1 = 1$, we have

$$\partial_t \phi = R^{(2)} + \nabla_1 \cdot \beta c_s^2 \tilde{\mathbf{S}}^{-1} \nabla_1 \cdot \mathbf{D} + \nabla_1 \cdot \mathbf{RH}^{(1)}, \quad (50)$$

where

$$\begin{aligned} \mathbf{RH}^{(1)} &= \tilde{\mathbf{S}}^{-1} [\partial_t \mathbf{B} + \nabla_1 \cdot \mathbf{C} - \mathbf{M}_{1,\tilde{G}}^{(1)} - \mathbf{M}_{1,F}^{(1)}] \\ &+ \frac{h}{2} [(\gamma_2 - 1) \mathbf{M}_{1,\tilde{G}}^{(1)} + (\gamma_1 - 1) \mathbf{M}_{1,F}^{(1)}]. \end{aligned} \quad (51)$$

If we take $\mathbf{RH} = 0$ and $\mathbf{K} = \beta c_s^2 \tilde{\mathbf{S}}^{-1}$, then Eq. (50) can be rewritten as Eq. (36). According to the results at ε and ε^2 scales, i.e., Eqs. (47a) and (36), Eq. (2) can be recovered exactly.

There are also two special cases when $\mathbf{RH} = 0$.

Scheme A: $\mathbf{B} = \mathbf{B}(\phi)$, $\mathbf{M}_{1,F} = 0$, $\gamma_1 = \gamma_2 = 0$. With these conditions, we have

$$\mathbf{RH} = \tilde{\mathbf{S}}^{-1} (\partial_t \mathbf{B} + \nabla \cdot \mathbf{C} - \mathbf{M}_{1,\tilde{G}}) - \frac{h}{2} \mathbf{M}_{1,\tilde{G}}. \quad (52)$$

When $\mathbf{RH} = 0$, one can obtain

$$\mathbf{M}_{1,\tilde{G}} = \left(\mathbf{I} + \frac{h}{2} \tilde{\mathbf{S}} \right)^{-1} (\partial_t \mathbf{B} + \nabla \cdot \mathbf{C}). \quad (53)$$

If we set

$$\mathbf{C}_{\alpha\beta} = \int \mathbf{B}'_{\alpha} \mathbf{B}'_{\beta} d\phi, \quad \text{or} \quad \frac{d\mathbf{C}_{\alpha\beta}}{d\phi} = \frac{d\mathbf{B}_{\alpha}}{d\phi} \frac{d\mathbf{B}_{\beta}}{d\phi}, \quad (54)$$

then one can derive the following equation:

$$\partial_t \mathbf{B} + \nabla_1 \cdot \mathbf{C} = \mathbf{B}' (\partial_t \phi + \nabla_1 \cdot \mathbf{B}) = \mathbf{B}' R^{(1)}. \quad (55)$$

Substituting Eq. (55) into Eq. (53) yields

$$\mathbf{M}_{1,\tilde{G}} = \left(\mathbf{I} + \frac{h}{2} \tilde{\mathbf{S}} \right)^{-1} \mathbf{B}' R. \quad (56)$$

Scheme B: $\mathbf{C} = 0$, $\mathbf{M}_{1,F} = 0$, $\gamma_1 = \gamma_2 = 0$. Under these conditions, we can obtain

$$\mathbf{RH} = \tilde{\mathbf{S}}^{-1} (\partial_t \mathbf{B} - \mathbf{M}_{1,\tilde{G}}) - \frac{h}{2} \mathbf{M}_{1,\tilde{G}}. \quad (57)$$

When $\mathbf{RH} = 0$, one can determine $\mathbf{M}_{1,\tilde{G}}$ as

$$\mathbf{M}_{1,\tilde{G}} = \left(\mathbf{I} + \frac{h}{2} \tilde{\mathbf{S}} \right)^{-1} \partial_t \mathbf{B}. \quad (58)$$

In our simulations, the term $\partial_t \mathbf{B}$ in Eqs. (58) and (43) can be estimated by $[\mathbf{B}(\mathbf{x}, t) - \mathbf{B}(\mathbf{x}, t - \Delta t)]/\Delta t$.

Finally we give some remarks on the present MRT-FDLBM.

Remark 1: It should be noted that the moment conditions of the source term G are different in two evolution processes. In

the main evolution Eq. (2), the coefficient matrix of G equals to \mathbf{I} , while it is $(\mathbf{I} + \frac{h}{2} \tilde{\mathbf{S}})^{-1}$ in the second evolution Eq. (7).

Remark 2: It can be found the values of $\bar{\theta}$ and θ_2 do not affect the CE analysis process, and thus we can take $\bar{\theta} = \theta_2 = \gamma_1 = \gamma_2 = 0$ and $\theta = \theta_1 = 1$ for simplicity. Besides, as we can see in Eq. (10) and Eq. (16), there is a Δt in the coefficient of ∂F . Thus, a first-order approximation in time is applied to compute ∂F for both scheme A and scheme B. Here, we can use $[F_j(\mathbf{x}, t) - F_j(\mathbf{x}, t - \Delta t)]/\Delta t$ to estimate $\partial_t F_j(\mathbf{x}, t)$ in Eq. (2), and use $[F_j(\mathbf{x} - \mathbf{c}_j h, t) - F_j(\mathbf{x} - \mathbf{c}_j h, t - \Delta t)]/\Delta t$ to estimate $\partial_t F_j(\mathbf{x} - \mathbf{c}_j h, t)$ in Eq. (7).

Remark 3: We point out that the collision matrices of the two evolution equations can be different from each other. That is, in addition to the relaxation parameters related to the diffusion coefficient \mathbf{K} , the remaining relaxation parameters in the two relaxation matrices can take different values.

IV. STABILITY ANALYSIS OF THE MRT-FDLBM

The linear stability analysis is a common tool to evaluate and improve the numerical stability of LBMs, whatever the collision model considered [47–49]. In this section, the von Neumann method [50] is used to analyze the numerical stability of the MRT-FDLBM, and for simplicity, the source term is neglected. We first write the main evolution Eq. (9) as

$$\begin{aligned} f_j(\mathbf{x}, t + \Delta t) &+ 0.5 \omega_{jk} [f_k(\mathbf{x}, t + \Delta t) - f_k^{\text{eq}}(\mathbf{x}, t + \Delta t)] \\ &= f_j(\mathbf{x}, t) - \Delta t \mathbf{c}_j \cdot \nabla f_j \left(\mathbf{x}, t + \frac{\Delta t}{2} \right) \\ &- 0.5 \omega_{jk} [f_k(\mathbf{x}, t) - f_k^{\text{eq}}(\mathbf{x}, t)], \end{aligned} \quad (59)$$

then to conduct a linear stability analysis, $f_j(\mathbf{x}, t)$ is expanded as

$$f_j(\mathbf{x}, t) = \overline{f_j^{\text{eq}}(\mathbf{x}, t)} + f'_j(\mathbf{x}, t), \quad (60)$$

where $\overline{f_j^{\text{eq}}(\mathbf{x}, t)}$ represents the global equilibrium distribution, and only depends on the mean value of ϕ that does not vary with time and space. $f'_j(\mathbf{x}, t)$ is the fluctuating quantity of $f_j(\mathbf{x}, t)$. With the help of Eq. (60), one can rewrite Eq. (59) as

$$\begin{aligned} &\left[\left(1 + \frac{\omega_{jk}}{2} \right) \delta_{jk} + \frac{\omega_{jk}}{2} \Gamma_{jk} \right] f'_j(\mathbf{x}, t + \Delta t) \\ &= \left[\left(1 - \frac{\omega_{jk}}{2} \right) \delta_{jk} + \frac{\omega_{jk}}{2} \Gamma_{jk} \right] f'_j(\mathbf{x}, t) \\ &- \delta_{jk} \Delta t \mathbf{c}_k \cdot \nabla f'_j \left(\mathbf{x}, t + \frac{\Delta t}{2} \right), \end{aligned} \quad (61)$$

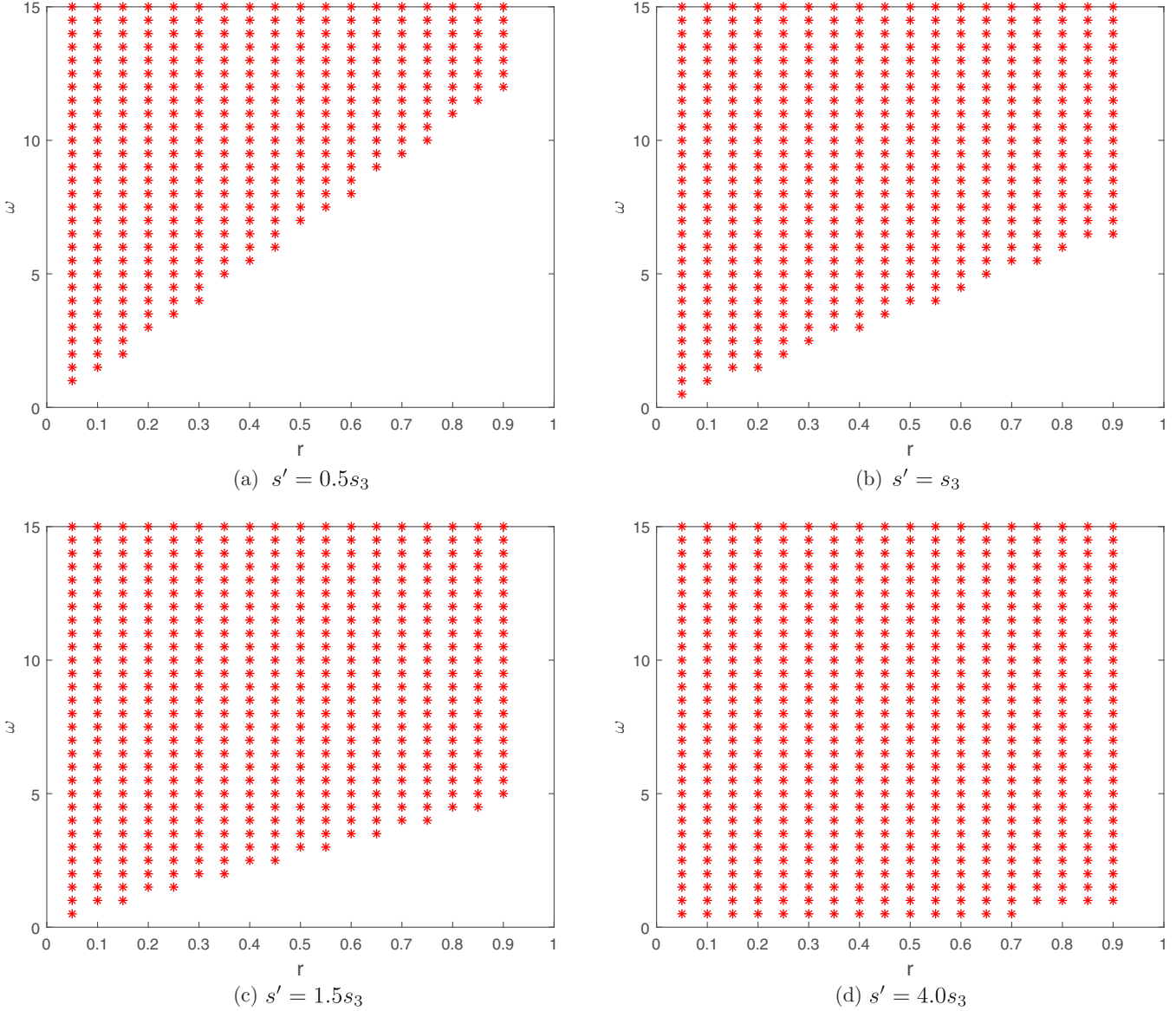


FIG. 2. Stability regions of the MRT-FDLBM with $u = (2.5, 2.5)^T$, $c = 5.0$.

where $\Gamma_{jk} = \partial f_j^{\text{eq}}(\mathbf{x}, t) / \partial f_k(\mathbf{x}, t)$. With the Fourier transform, one can also get

$$\begin{pmatrix} \mathbf{I} + 0.5\boldsymbol{\omega} - 0.5\boldsymbol{\omega}\boldsymbol{\Gamma} & 0 \\ 0 & \mathbf{I} \end{pmatrix} \begin{bmatrix} F'_k(\boldsymbol{\kappa}, t + \Delta t) \\ F'_k(\boldsymbol{\kappa}, t + \frac{1}{2}\Delta t) \end{bmatrix} = \begin{pmatrix} -r\mathbf{T} & \mathbf{I} - 0.5\boldsymbol{\omega} + 0.5\boldsymbol{\omega}\boldsymbol{\Gamma} \\ \mathbf{I} & 0 \end{pmatrix} \begin{bmatrix} F'_k(\boldsymbol{\kappa}, t + \frac{1}{2}\Delta t) \\ F'_k(\boldsymbol{\kappa}, t) \end{bmatrix}, \tag{62}$$

where $F'_k(\boldsymbol{\kappa}, t) = \int f'_k(\mathbf{x}, t) \exp(-i\boldsymbol{\kappa} \cdot \mathbf{x}) d\mathbf{x}$ with $\boldsymbol{\kappa} = (\kappa_x, \kappa_y)$ being the wave number. From Eq. (62), we can determine the growth matrix $\hat{\mathbf{G}}$ as

$$\hat{\mathbf{G}} = \begin{pmatrix} \mathbf{I} + 0.5\boldsymbol{\omega} - 0.5\boldsymbol{\omega}\boldsymbol{\Gamma} & 0 \\ 0 & \mathbf{I} \end{pmatrix}^{-1} \begin{pmatrix} -r\mathbf{T} & \mathbf{I} - 0.5\boldsymbol{\omega} + 0.5\boldsymbol{\omega}\boldsymbol{\Gamma} \\ \mathbf{I} & 0 \end{pmatrix} \tag{63}$$

where $r = \Delta t / \Delta x$ and $\mathbf{T} = \text{diag}(T_0, T_1, \dots, T_q)$. For the mixed difference scheme,

$$T_j = i(1 - \eta)(\sin \vartheta_{jx} + \sin \vartheta_{jy}) + \frac{\eta}{2}[6 - 4 \exp(-\vartheta_{jx}) - 4 \exp(-\vartheta_{jy}) + \exp(-2i\vartheta_{jx}) + \exp(-2i\vartheta_{jy})], \tag{64}$$

where $\vartheta_{jx} = \kappa_x c_{jx} \Delta \chi$ and $\vartheta_{jy} = \kappa_y c_{jy} \Delta \chi$.

According to the von Neumann stability condition, we need to obtain the spectral radius of the matrix $\hat{\mathbf{G}}$. However,

for the general NCDE, there are too many free parameters in the matrix $\hat{\mathbf{G}}$, and it is inconvenient to discuss the spectral

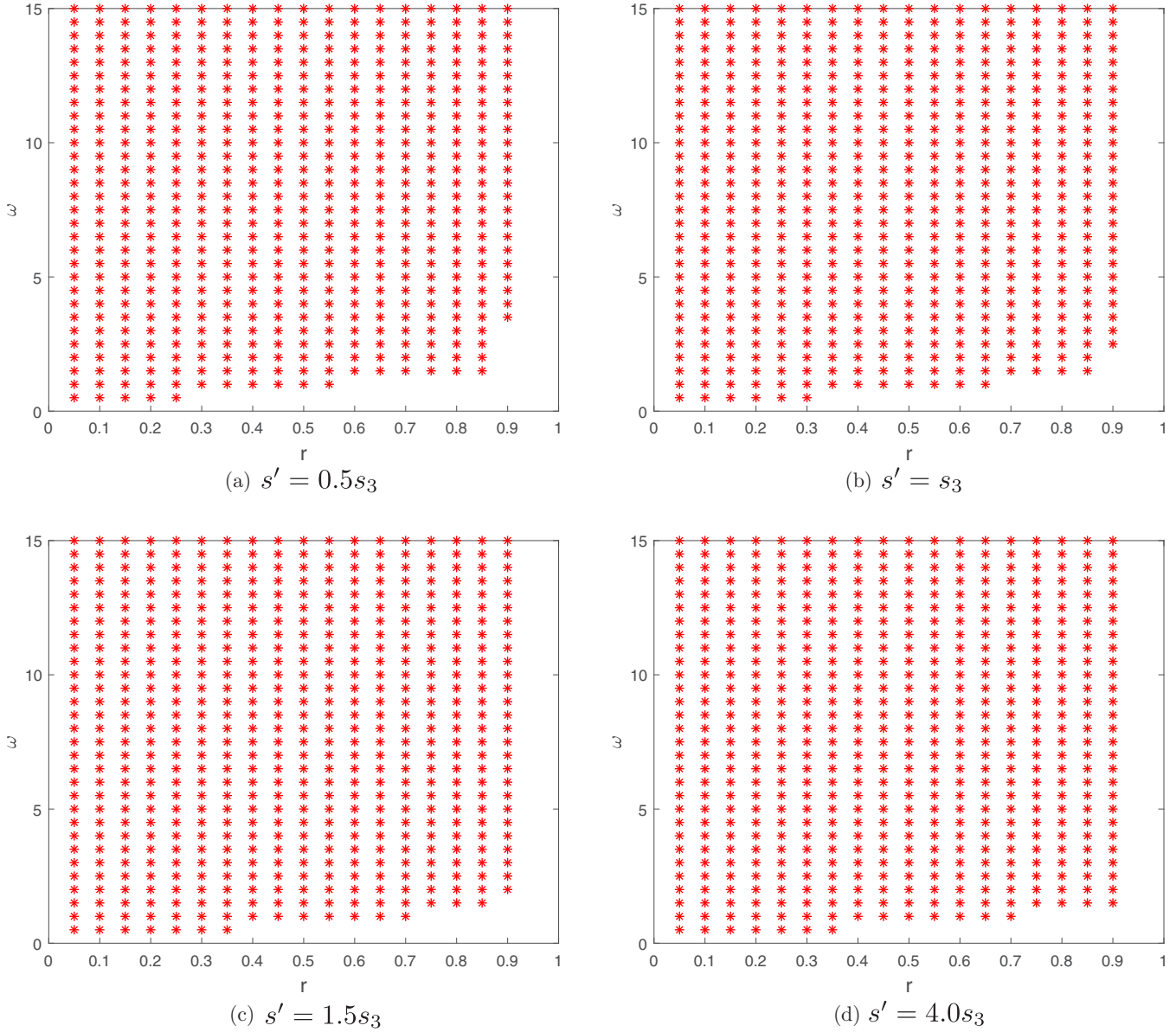


FIG. 3. Stability regions of the MRT-FDLBM with $s_5 = 0.5s_3$ and $\mathbf{u} = (2.5, 2.5)^T$, $c = 5.0$.

radius of \hat{G} . Therefore, in the following, we only consider a linear convection-diffusion problem to analyze the numerical stability of the MRT-FDLBM. Here we take D2Q9 lattice model as an example and set $\mathbf{u} = (2.5, 2.5)^T$, $c = 5.0$. The spectral radius of the matrix \hat{G} is the function of ω and r with $0 \leq \kappa_x \Delta x, \kappa_y \Delta y \leq \pi$. To simplify the analysis on the stable domain of the MRT-FDLBM, we set the relaxation parameters as $s_0 = s_1 = s_2 = s_4 = s_6 = s_7 = s_8 = s'$ and $s_3 = s_5$. Figure 2 shows some stability regions of the MRT-FDLBM with the relaxation parameter $s' = 0.5s_3, s_3, 1.5s_3$, and $4.0s_3$. From this figure, one can find that the stability region of MRT-FDLBM increases with the relaxation parameter s' . Besides, we also conduct the stability analysis of MRT-FDLBM for the anisotropic convection-diffusion equation. Figure 3 shows the stability region with $s_5 = 0.5s_3$. The stability region increases slightly as s' increases. And Fig. 4 presents the stability region with $s' = 1.5s_3$. It can be found that the stability region increases significantly as s_5 increases. At last, we test the sta-

bility of MRT-FDLBM with $\mathbf{u} = (0.5, 0.5)^T$ and $c = 1.0$. The results are shown in Fig. 5, it can be found that the stability region increases slightly as s' increases. This phenomenon is similar to Fig. 2. It is interesting to see that a similar behavior is obtained with the LBM [51], and this common trick can improve the stability of LBMs in under-resolved conditions whatever the targeted physics [52].

To numerically validate above analysis on the stability region of the MRT-FDLBM, we consider a linear convection-diffusion problem

$$\partial_t \phi + \nabla \cdot (\phi \mathbf{u}) = \nabla \cdot \kappa \nabla \phi, \tag{65}$$

with the analytical solution

$$\phi(\mathbf{x}, t) = \frac{\mu_0 \delta_0^2}{\delta_0^2 + 2\kappa t} \exp \left[-\frac{(x - u_x t)^2 + (y - u_y t)^2}{2(\delta_0^2 + 2\kappa t)} \right], \tag{66}$$

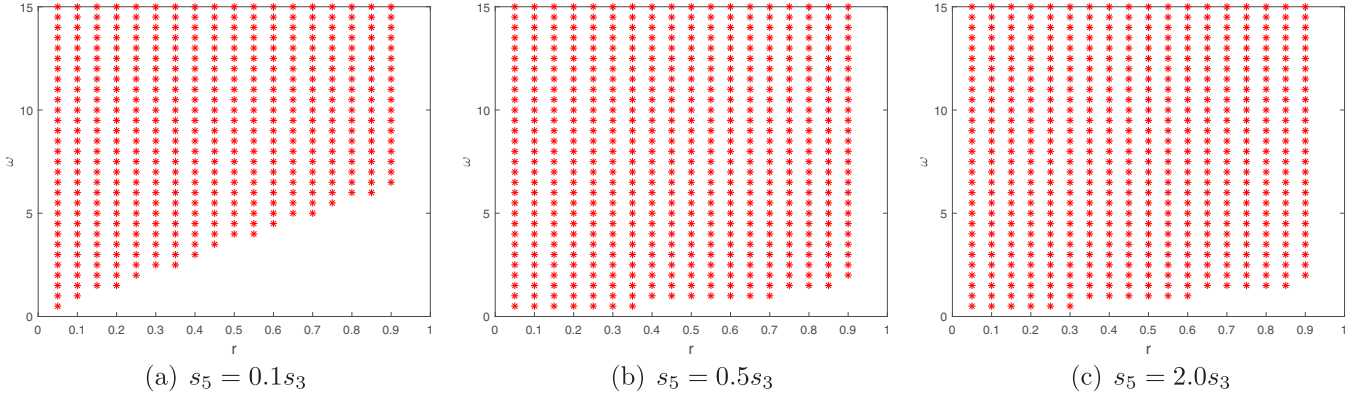


FIG. 4. Stability regions of the MRT-FDLBM with $s' = 1.5s_3$ and $u = (2.5, 2.5)^T$, $c = 5.0$.

where κ is the diffusion coefficient, $\mu_0 = 100$, and $\delta_0 = 20.0$. The computational domain is $[0, 1] \times [0, 1]$ with $\Delta x = 0.01$ and $c = 5$. As we can see from Fig. 2, the point $(\omega, r) = (0.15, 1.0)$ is out of the stability region when $s' = 0.5s_3$ and $s' = s_3$, and it is inside the stable region when $s' = 1.5s_3$ or

$s' = 4.0s_3$. Here we take $\omega = 0.15$ and $r = 1.0$ in our simulations and plot the results in Fig. 6. From this figure, we can observe that MRT-FDLBMs with $s' = 0.5s_3$ and $s' = s_3$ are unstable, while the MRT-FDLBMs with $s' = 1.5s_3$ and $s' = 4.0s_3$ work well, which is consistent with our above

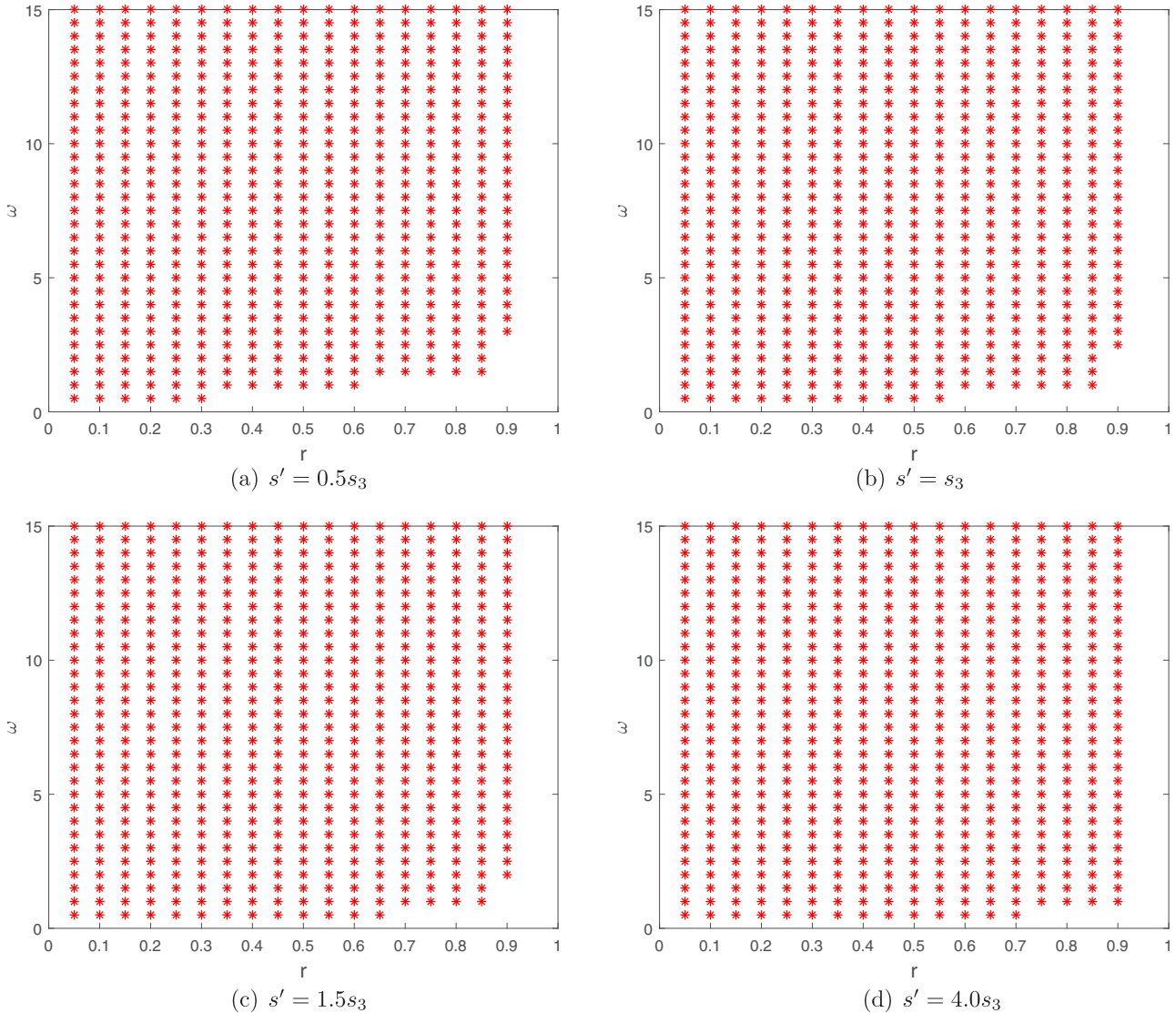


FIG. 5. Stability regions of the MRT-FDLBM with $u = (0.5, 0.25)^T$ and $c = 1.0$.

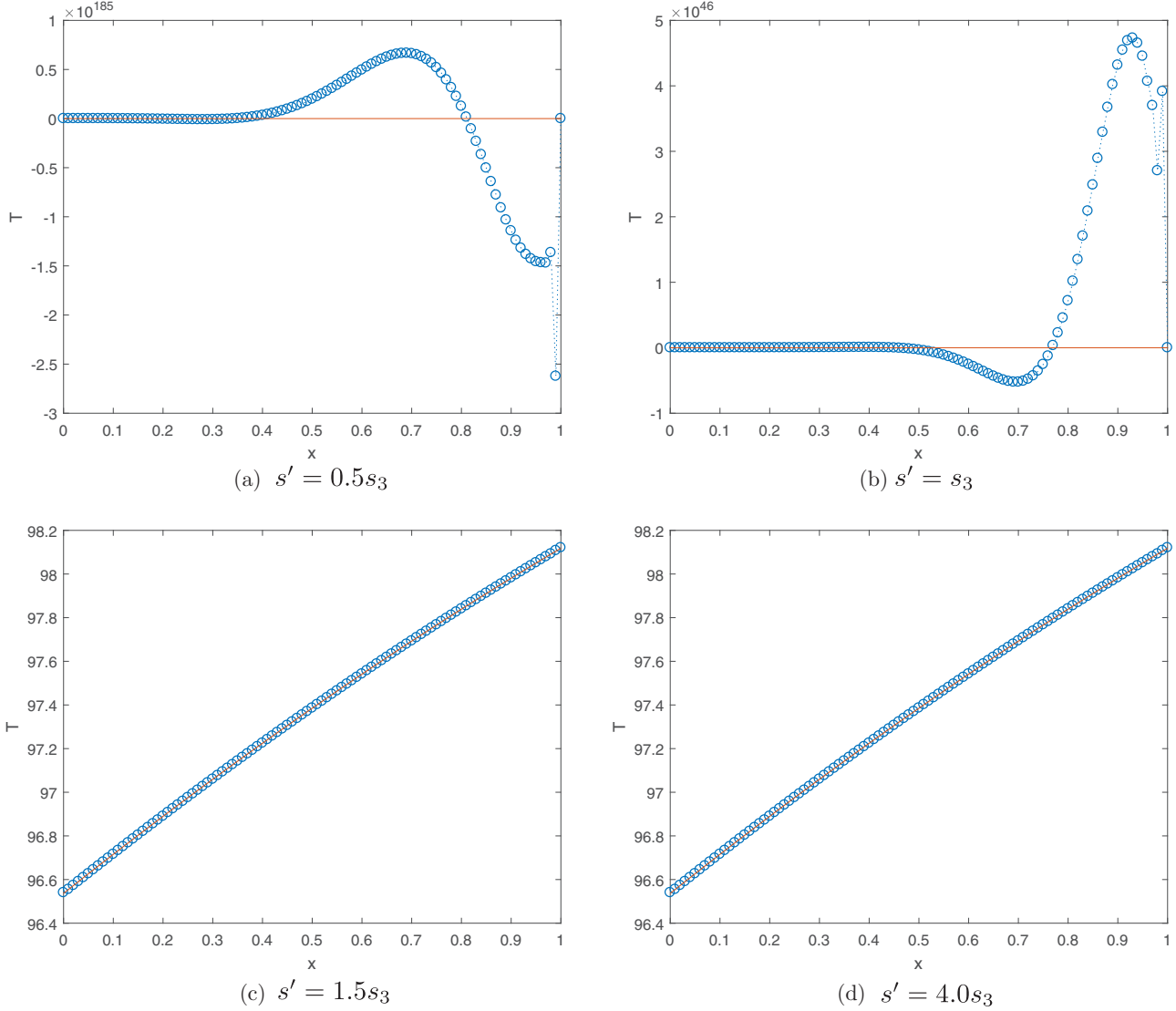


FIG. 6. Distributions of the scalar variable ϕ with different MRT-FDLBMs ($t = 1.5, u = (2.5, 2.5)^T, c = 5.0$).

analysis. In addition, it should be noted that when $s' = s_3$, the MRT-FDLBM would reduce to the SRT-FDLBM, and the above results also indicate that the MRT-FDLBM with an appropriate relaxation matrix is more stable than the SRT-FDLBM.

V. MODEL SIMPLIFICATION

In general, the MRT model is more stable than the BGK model, while the BGK model is more efficient than the MRT

model. The present MRT-FDLBM is obviously more stable than SRT-FDLBM, as shown in the previous section, but the matrix calculation in the MRT-FDLBM will consume more CPU time. To improve the computational efficiency of MRT-FDLBM, a simplified MRT-FDLBM (SMRT-FDLBM) is developed in this section. To this end, we can adopt a BGK model to deal with the collision term in Eq. (7), while the collision term in the Eq. (2) is still approximated by the MRT model. Following this way, Eq. (7) can be expressed as

$$f_j(\mathbf{x}, t+h) - f_j(\mathbf{x} - \mathbf{c}_j h, t) = \frac{h}{2} \left[-\frac{1}{\tau} f_j^{ne}(\mathbf{x}, t+h) - \frac{1}{\tau} f_k^{ne}(\mathbf{x} - \mathbf{c}_j h, t) \right] + h \left(F_j + \bar{G}_j + \frac{h}{2} \bar{D}_j F_j + \frac{h}{2} \bar{\bar{D}}_j \bar{G}_j \right) (\mathbf{x} - \mathbf{c}_j h, t), \quad (67)$$

which can also be written into an explicit form,

$$\begin{aligned} \bar{f}_j(\mathbf{x}, t+h) &= \bar{f}_j^+(\mathbf{x} - \mathbf{c}_j h, t) + h \left[F_j + \bar{G}_j + \frac{h}{2} \bar{D}_j F_j + \frac{h}{2} \bar{\bar{D}}_j \bar{G}_j \right] (\mathbf{x} - h \mathbf{c}_j, t) \\ &= \bar{f}_j^+(\mathbf{x}, t) - \mathbf{c}_j h \cdot \nabla \bar{f}_j^+(\mathbf{x}, t) + h \left[F_j + \bar{G}_j + \frac{h}{2} \bar{D}_j F_j + \frac{h}{2} \bar{\bar{D}}_j \bar{G}_j \right] (\mathbf{x}, t) + O(h^2). \end{aligned} \quad (68)$$

Besides, if we neglect the terms $\bar{D}_j F_j$ and $\bar{D}_j \bar{G}_j$ due to the fact that the coefficients of this two terms are of order $O(h^2)$, then Eq. (68) can be simplified as

$$\begin{aligned} \bar{f}_j(\mathbf{x}, t+h) &= \bar{f}_j^+(\mathbf{x}, t) - \mathbf{c}_j h \cdot \nabla \bar{f}_j^+(\mathbf{x}, t) \\ &\quad + h[F_j + \bar{G}_j](\mathbf{x}, t) + O(h^2), \end{aligned} \quad (69)$$

where

$$\bar{f} = \frac{2\tau + h}{2\tau} f - \frac{h}{2\tau} f^{\text{eq}}, \quad (70a)$$

$$\bar{f}^+ = \frac{2\tau - h}{2\tau} f + \frac{h}{2\tau} f^{\text{eq}}. \quad (70b)$$

With the help of Eq. (13a), we can derive

$$\bar{f}^+ = \frac{2\tau - h}{2\tau} (\mathbf{I} + h\tilde{\Lambda})^{-1} \hat{f} + (\mathbf{I} + h\tilde{\Lambda})^{-1} \left(h\tilde{\Lambda} + \frac{h\mathbf{I}}{2\tau} \right) f^{\text{eq}}. \quad (71)$$

The implementation process of the SMRT-FDLBM is similar to that in Fig. 1. The relationship between τ and the diffusion coefficient K is $c_s^2 \tau \mathbf{I} = \mathbf{K}$, and the relationship between τ and \tilde{S} is $\frac{1}{\tau} \mathbf{I} = \tilde{S}$.

Remark: Referring to the TRT-LBM models [27,53], two-relaxation-times FDLBM (TRT-FDLBM), as a special case of MRT-FDLBM, can also improve the computational efficiency. The difference between the two model is mainly reflected in the collision term. The calculation process of the TRT-FDLBM is presented in Appendix B.

VI. THE MRT-FDLBM FOR COMPLEX-VALUED NCDE

In the previous CE analysis, we can find that both the real- and complex-valued NCDEs can be recovered from the MRT-FDLBM. In other words, the method can also be applied to solve the complex-valued NCDE, and the implementation details would be discussed in this section. With the following complex variables:

$$\begin{aligned} \mathbf{f} &= \mathbf{f}_r + i\mathbf{f}_c, \quad \mathbf{f}^{\text{eq}} = \mathbf{f}_r^{\text{eq}} + i\mathbf{f}_c^{\text{eq}}, \quad \mathbf{F} = \mathbf{F}_r + i\mathbf{F}_c, \\ \mathbf{G} &= \mathbf{G}_r + i\mathbf{G}_c, \quad \mathbf{\Lambda} = \mathbf{\Lambda}_r + i\mathbf{\Lambda}_c, \end{aligned} \quad (72)$$

where $i^2 = -1$. We can write Eq. (10) as

$$\bar{N}_1(\mathbf{x}, t + \Delta t) = \bar{N}_1^+(\mathbf{x}, t) - \Delta t \mathbf{c} \cdot \mathbf{f}_r(\mathbf{x}, t + h) + \Delta t \left(\mathbf{F}_r + \mathbf{G}_r + \frac{\Delta t}{2} \theta \partial_t \mathbf{F}_r + \frac{\Delta t}{2} \bar{\theta} \partial_t \mathbf{G}_r \right)(\mathbf{x}, t), \quad (73)$$

$$\bar{N}_2(\mathbf{x}, t + \Delta t) = \bar{N}_2^+(\mathbf{x}, t) - \Delta t \mathbf{c} \cdot \mathbf{f}_c(\mathbf{x}, t + h) + \Delta t \left(\mathbf{F}_c + \mathbf{G}_c + \frac{\Delta t}{2} \theta \partial_t \mathbf{F}_c + \frac{\Delta t}{2} \bar{\theta} \partial_t \mathbf{G}_c \right)(\mathbf{x}, t), \quad (74)$$

where $\theta = 1$, $\bar{\theta} = 0$,

$$\bar{N}_1 = \mathbf{f}_r + \frac{1}{2} [\Delta t \tilde{\Lambda}_r (\mathbf{f}_r - \mathbf{f}_r^{\text{eq}}) - \Delta t \tilde{\Lambda}_c (\mathbf{f}_c - \mathbf{f}_c^{\text{eq}})], \quad (75a)$$

$$\bar{N}_2 = \mathbf{f}_c + \frac{1}{2} [\Delta t \tilde{\Lambda}_r (\mathbf{f}_c - \mathbf{f}_c^{\text{eq}}) + \Delta t \tilde{\Lambda}_c (\mathbf{f}_r - \mathbf{f}_r^{\text{eq}})], \quad (75b)$$

and

$$\bar{N}_1^+ = \mathbf{f}_r - \frac{1}{2} [\Delta t \tilde{\Lambda}_r (\mathbf{f}_r - \mathbf{f}_r^{\text{eq}}) - \Delta t \tilde{\Lambda}_c (\mathbf{f}_c - \mathbf{f}_c^{\text{eq}})], \quad (76a)$$

$$\bar{N}_2^+ = \mathbf{f}_c - \frac{1}{2} [\Delta t \tilde{\Lambda}_r (\mathbf{f}_c - \mathbf{f}_c^{\text{eq}}) + \Delta t \tilde{\Lambda}_c (\mathbf{f}_r - \mathbf{f}_r^{\text{eq}})]. \quad (76b)$$

Similarly, the evolution equation Eq. (7) is also decomposed of two parts,

$$\bar{N}_1(\mathbf{x}, t+h) = \bar{N}_1^+(\mathbf{x}, t) - h\mathbf{c} \cdot \nabla \bar{N}_1^+(\mathbf{x}, t) + h \left(\mathbf{F}_r + \bar{G}_r + \frac{h}{2} \bar{D} \mathbf{F}_r + \frac{h}{2} \bar{D} \bar{G}_r \right)(\mathbf{x}, t) + O(h^2), \quad (77a)$$

$$\bar{N}_2(\mathbf{x}, t+h) = \bar{N}_2^+(\mathbf{x}, t) - h\mathbf{c} \cdot \nabla \bar{N}_2^+(\mathbf{x}, t) + h \left(\mathbf{F}_c + \bar{G}_c + \frac{h}{2} \bar{D} \mathbf{F}_c + \frac{h}{2} \bar{D} \bar{G}_c \right)(\mathbf{x}, t) + O(h^2). \quad (77b)$$

Let $\bar{\omega}_r = h\tilde{\Lambda}_r$, we have

$$\bar{N}_1 = \mathbf{f}_r + \frac{1}{2} [\bar{\omega}_r (\mathbf{f}_r - \mathbf{f}_r^{\text{eq}}) - \bar{\omega}_c (\mathbf{f}_c - \mathbf{f}_c^{\text{eq}})], \quad (78a)$$

$$\bar{N}_2 = \mathbf{f}_c + \frac{1}{2} [\bar{\omega}_r (\mathbf{f}_c - \mathbf{f}_c^{\text{eq}}) + \bar{\omega}_c (\mathbf{f}_r - \mathbf{f}_r^{\text{eq}})], \quad (78b)$$

and

$$\bar{N}_1^+ = \mathbf{f}_r - \frac{1}{2} [\bar{\omega}_r (\mathbf{f}_r - \mathbf{f}_r^{\text{eq}}) - \bar{\omega}_c (\mathbf{f}_c - \mathbf{f}_c^{\text{eq}})], \quad (79a)$$

$$\bar{N}_2^+ = \mathbf{f}_c - \frac{1}{2} [\bar{\omega}_r (\mathbf{f}_c - \mathbf{f}_c^{\text{eq}}) + \bar{\omega}_c (\mathbf{f}_r - \mathbf{f}_r^{\text{eq}})]. \quad (79b)$$

In Fig. 7, we present the implementation process of the MRT-FDLBM for complex-valued NCDE.

VII. NUMERICAL SIMULATION

In this section, five different numerical examples are used to test the present MRT-FDLBM, including the isotropic convection-diffusion equation, the Burgers-Fisher equation, the nonlinear heat conduction equation, the sine-Gordon equation and the nonlinear Schrödinger equation. Unless otherwise specified, the distribution function is initialized by the equilibrium distribution function, i.e., $\hat{f}_j(\mathbf{x}, t) = \hat{f}_j^{\text{eq}}(\mathbf{x}, t)$. The boundary conditions are treated by nonequilibrium extrapo-

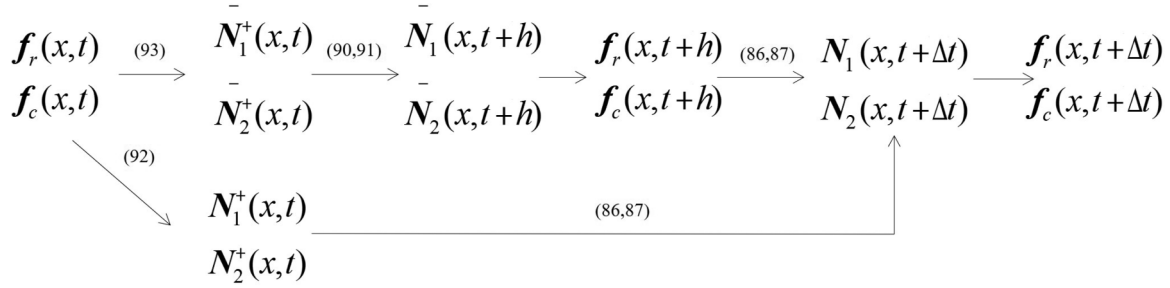


FIG. 7. The implementation process of the MRT-FDLBM for complex-valued NCDE.

lation scheme [54], and the D2Q9 lattice model is used. To test the accuracy of the MRT-FDLBM, the following global relative errors (GRE) is adopted:

$$GRE = \frac{\sum_i |\phi(x_i, t) - \phi^*(x_i, t)|}{\sum_i |\phi^*(x_i, t)|}, \quad (80)$$

where $\phi(x, t)$ and $\phi^*(x, t)$ denote numerical and analytical solutions, respectively. In addition, to give a comparison of different models, the previous FDLBM [55] is marked as FDLBM, the MRT-FDLBM with $s_i = s_3$, ($0 \leq i \leq 8$) is recorded as SRT-FDLBM, and MRT-FDLBM with $s_0 = s_1 = s_2 = s_4 = s_6 = s_7 = s_8 = 1.5s_3$ is denoted as MRT-FDLBM. From Sec. IV, it can be found the value $s' = 4.0s_3$ is a more stable case. However, usually an increased stability implies a lower accuracy. Hence, considered stability and accuracy synthetically, we take $s' = 1.5s_3$ for all MRT-FDLBM in this section.

A. Example 1: Isotropic convection-diffusion equation

The following two-dimensional isotropic CDE is first considered to test the accuracy of the MRT-FDLBM,

$$\partial_t \phi + u_x \partial_x \phi + u_y \partial_y \phi = \kappa (\partial_{xx} \phi + \partial_{yy} \phi) + R, \quad (81)$$

where $u_x = u_y = 0.1$, κ is the diffusion coefficient, and the source term R is given by

$$R = \exp[(1 - 2\pi^2 \kappa)t] \sin[\pi(x + y)] + \pi(u_x + u_y) \cos[\pi(x + y)]. \quad (82)$$

The analytical solution of this problem can be expressed as

$$\phi(x, y, t) = \exp[(1 - 2\pi^2 \kappa)t] \sin[\pi(x + y)]. \quad (83)$$

For this problem, two schemes of the MRT-FDLBM are considered. In scheme A, the functions \mathbf{B} , \mathbf{C} , and \mathbf{D} should be set as $\mathbf{B} = \phi \mathbf{u}$, $\mathbf{C} = \phi \mathbf{u} \mathbf{u}$, and $\mathbf{D} = \phi \mathbf{I}$, while in scheme B, $\mathbf{B} = \phi \mathbf{u}$, $\mathbf{C} = \mathbf{0}$, and $\mathbf{D} = \phi \mathbf{I}$.

In the following simulations, the lattice size is 201×201 , the periodic boundary condition is adopted for all boundaries, and the initial condition is given by the analytical solution at $t = 0$. Figure 8 shows the results of two schemes at $Pe = 1000$ ($Pe = Lu_x/\kappa$, the characteristic length $L = 2.0$). From this figure, it can be seen that the numerical results of two schemes at different time agree well with the analytical solution. In addition, this problem is also used to test the convergence rates of the MRT-FDLBMs, including the MRT-FDLBM with scheme A, the MRT-FDLBM with scheme B, and the SMRT-FDLBM with scheme B. For this purpose, the time step is varied from 0.001 to 0.005 to test the convergence rate in time, and the lattice spacing changes from 2.0/200 to

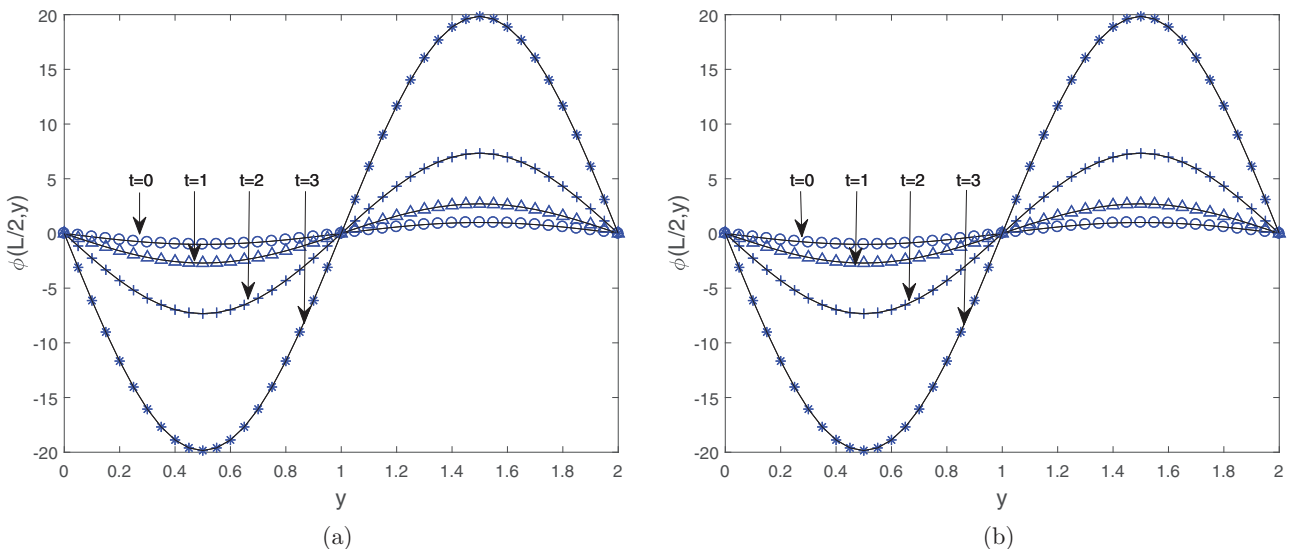


FIG. 8. Example 1: Profiles of scalar variable ϕ at $Pe = 1000$. (a) Scheme A. (b) Scheme B.

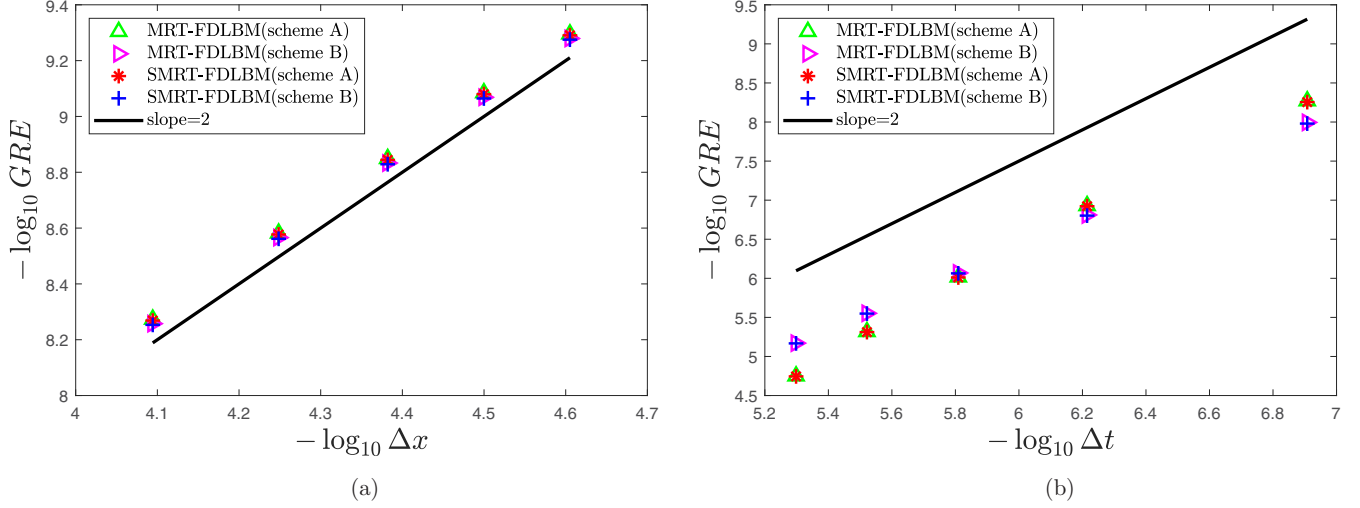


FIG. 9. Example 1: GREs of the MRT-FDLBM with scheme A and scheme B and SMRT-FDLBM with scheme A and scheme B. (a) GREs at different grid sizes and (b) GREs at different time.

2.0/120 to test the convergence rate in space. And the CFL condition number is fixed to 0.1 As shown in Fig. 9, both the MRT-FDLBM and SMRT-FDLBM have the second-order convergence rates in time and space, which is consistent with the theoretical analysis. Finally, we also conducted a comparison among different LBMs and present the results in Table I. From this table, one can find that the results of MRT-FDLBM and SRT-FDLBM are more accurate than those of standard LBM and FDLBM.

B. Example 2: Two-dimensional Burgers-Fisher equation

We continued to consider the two-dimensional Burgers-Fisher equation with a source term,

$$\partial_t \phi + a\phi^\delta \partial_x \phi - b(\partial_{xx} \phi + \partial_{yy} \phi) = \kappa \phi(1 - \phi^\delta). \quad (84)$$

Under the proper initial and boundary conditions, one can obtain its analytical solution,

$$\phi(x, y, t) = \frac{1}{2} + \frac{1}{2} \tanh[A(x + y - \omega t)]^{\frac{1}{2}}, \quad (85)$$

where the parameters A and ω are given by

$$A = -\frac{a\delta}{4\kappa(\delta + 1)}, \quad \omega = \frac{a^2 + 2b\kappa(\delta + 1)}{a(\delta + 1)}. \quad (86)$$

In our simulations, $\kappa = 1.0$, $\delta = 2.0$, $\Delta x = 0.01$, $b = 0.05$, and $CFL = 0.1$. Scheme B with $\mathbf{B}(\phi) = \phi^{\delta+1}(\frac{a}{\delta+1}, 0)^T$, $\mathbf{C} =$

$\mathbf{0}$, and $\mathbf{D} = \phi \mathbf{I}$ is adopted for this problem. In Fig. 10, we present some results with the uniform mesh 300×300 and the nonuniform mesh 150×300 . As seen from this figure, the results of MRT-FDLBMs with the uniform mesh and nonuniform mesh are in good agreement with the analytical solution. However, we point out that when MRT-FDLBM with uniform grid is adopted, the GRE is 2.7775×10^{-4} and the CPU time is 698.03 s; if the MRT-FDLBM with the nonuniform grid is used, then the GRE is 3.2540×10^{-4} and the CPU time decreases to 300.81 s. These results show that it is feasible to use the nonuniform grids to improve computational efficiency of the MRT-FDLBM. In addition, we also measured the GREs at different grid sizes in Fig. 11. From this figure, one can observe that the MRT-FDLBMs with the uniform mesh and nonuniform mesh have the second-order convergence rate in space. We also carried out a comparison of four different LBMs, including LBM with the uniform grid, FDLBM with the nonuniform grid (FDLBM $'$), SRT-FDLBM with the nonuniform grid (SRT-FDLBM $'$), and MRT-FDLBM with the rectangular grid (MRT-FDLBM $'$). As shown in Table II, generally, the MRT-FDLBM $'$ is more accurate than the LBM, FDLBM $'$, and SRT-FDLBM $'$.

In order to visually show the computational efficiency of present MRT-FDLBM, we compared the three models, namely FDLBM, SRT-FDLBM, and MRT-FDLBM. In Table III, we set all three models to evolve 10 000 steps,

TABLE I. Example 1: A Comparison of GRE among LBM, FDLBM, and MRT-FDLBM.

Pe	Model	$c = 1$	$c = 10$	$c = 50$	$c = 100$
Pe = 100	LBM	7.0178×10^{-3}	1.2899×10^{-4}	2.2869×10^{-4}	2.3604×10^{-4}
	FDLBM	2.0227×10^{-3}	2.0837×10^{-4}	9.3423×10^{-5}	8.6576×10^{-5}
	SRT-FDLBM	6.1175×10^{-4}	8.4876×10^{-5}	5.1503×10^{-5}	5.0048×10^{-5}
	MRT-FDLBM	6.8405×10^{-4}	8.6859×10^{-5}	5.1637×10^{-5}	5.0029×10^{-5}
Pe = 10 000	LBM	3.0159×10^{-4}	1.9679×10^{-4}	2.0024×10^{-4}	2.0371×10^{-4}
	FDLBM	1.4896×10^{-3}	1.7425×10^{-4}	8.9458×10^{-4}	8.6282×10^{-5}
	SRT-FDLBM	1.1580×10^{-4}	5.0729×10^{-5}	5.0737×10^{-5}	5.2135×10^{-5}
	MRT-FDLBM	9.9619×10^{-5}	5.0717×10^{-5}	5.0737×10^{-5}	5.2134×10^{-5}

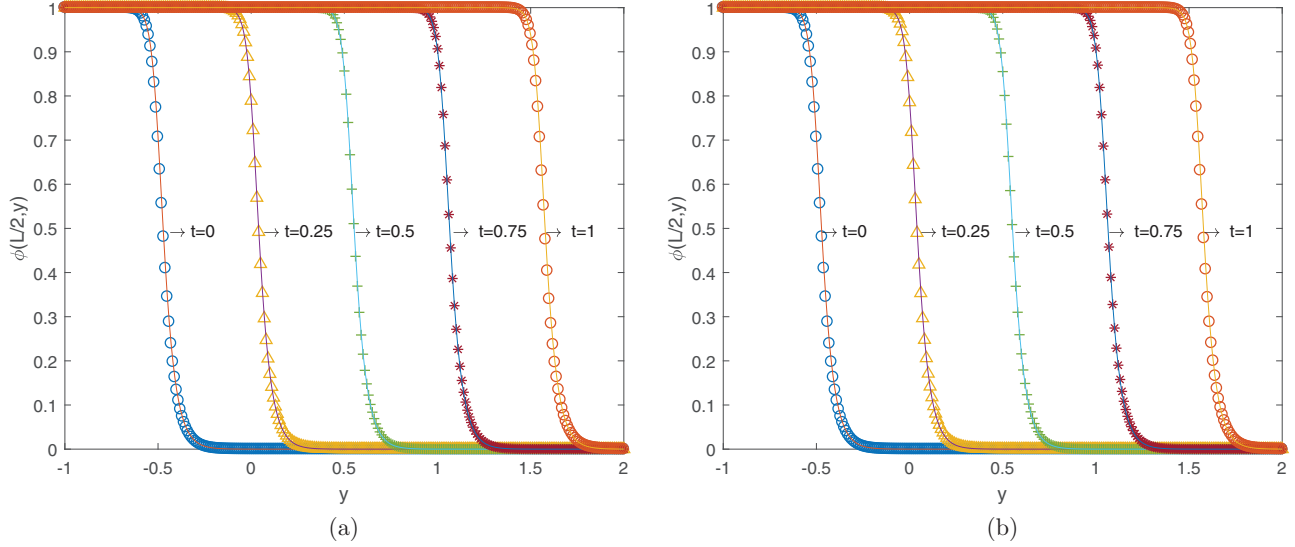


FIG. 10. Example 2: Profiles of scalar variable ϕ at different time [(a) uniform grid (300×300) and (b) nonuniform grid (150×300)].

and the results indicate that the single-step calculation time of MRT-FDLBM is 2.92 times as much as that of FDLBM. Besides, we also test the computational time when the GRE less than 10^{-5} . It can be seen from Table IV that the total calculation time of MRT-FDLBM is less than that of FDLBM. The results indicate that although the single-step evolution of MRT-FDLBM is more complex, the computational efficiency of MRT-FDLBM is not necessarily lower than that of FDLBM. The reason can be attributed to that the maximal CFL condition number of MRT-FDLBM is greater than that of FDLBM.

At last, the stability of the FDLBM, SRT-FDLBM, MRT-FDLBM, and SMRT-FDLBM are also tested with this example, and the results are presented in Table V. As we can see, SMRT-FDLBM and MRT-FDLBM work well, while FDLBM and SRT-FDLBM are unstable under the large CFL condition number and the large parameter b . More specially,

SMRT-FDLBM and MRT-FDLBM are more stable than SRT-FDLBM and FDLBM.

C. Example 3: Two-dimensional sine-Gordon equation

We now consider the two-dimensional sine-Gordon equation,

$$\partial_{tt}\phi = (\partial_{xx}\phi + \partial_{yy}\phi) - \sin(\phi). \tag{87}$$

The initial condition and boundary condition of this problem can be given by

$$\phi(x, y, 0) = p(x, y), \quad \partial_t\phi(x, y, 0) = q(x, y), \tag{88}$$

$$\partial_x\phi = 0, \quad x = \pm a; \quad \partial_y\phi = 0, \quad y = \pm b. \tag{89}$$

For the case of one circular ring soliton, the functions $p(x, y)$ and $q(x, y)$ can be expressed as

$$p(x, y) = \alpha \arctan \exp\{3 - \sqrt{x^2 + y^2}\}, \quad q(x, y) = 0, \tag{90}$$

$$-14 \leq x, y \leq 14,$$

while the two circular ring soliton, they should be given by

$$p(x, y) = \alpha \sum_{j=1}^2 \arctan \exp\{\gamma[4 - \sqrt{(x+x_j)^2 + (y+y_j)^2}]\},$$

$$q(x, y) = \beta \sum_{j=1}^2 \operatorname{sech}\{\gamma[4 - \sqrt{(x+x_j)^2 + (y+y_j)^2}]\}. \tag{91}$$

The computational domain of this case is fixed to be $-30 \leq x \leq 10, -21 \leq y \leq 7$, and $\{(x_j, y_j)\} = \{(3, 7), (17, 7)\}$ [18].

For the case of four circular ring soliton, the following standard conditions are adopted [18],

$$p(x, y) = \alpha \sum_{j=1}^4 \arctan \exp\{\gamma[4 - \sqrt{(x+x_j)^2 + (y+y_j)^2}]\},$$

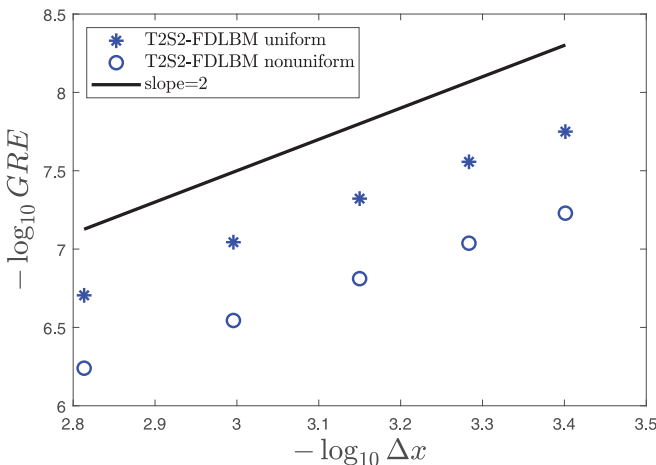


FIG. 11. Example 2: The GREs of the MRT-FDLBM with different grid sizes.

TABLE II. Example 2: The GREs of LBM, FDLBM^r, SRT-FDLBM^r, and MRT-FDLBM^r at $t = 1.0$.

b	Method	$c = 10$		$c = 20$	
		$a = 4$	$a = 6$	$a = 4$	$a = 6$
$b = 0.05$	LBM	9.4112×10^{-4}	9.7833×10^{-4}	2.3240×10^{-4}	2.3628×10^{-4}
	FDLBM ^r	1.0411×10^{-3}	1.0358×10^{-3}	3.8251×10^{-4}	3.3586×10^{-4}
	SRT-FDLBM ^r	7.9071×10^{-4}	7.0436×10^{-4}	4.2979×10^{-4}	3.8130×10^{-4}
	MRT-FDLBM ^r	3.9426×10^{-4}	3.2185×10^{-4}	3.0783×10^{-4}	2.2657×10^{-4}
$b = 0.1$	LBM	1.3549×10^{-3}	1.5879×10^{-3}	3.4558×10^{-4}	3.9627×10^{-4}
	FDLBM ^r	1.4368×10^{-3}	1.6554×10^{-3}	4.4683×10^{-4}	4.7877×10^{-4}
	SRT-FDLBM ^r	8.9130×10^{-4}	9.7677×10^{-4}	3.0855×10^{-4}	2.8205×10^{-4}
	MRT-FDLBM ^r	4.3223×10^{-4}	3.8305×10^{-4}	2.2312×10^{-4}	1.9187×10^{-4}

$$q(x, y) = \beta \sum_{j=1}^4 \operatorname{sech}\{\gamma[4 - \sqrt{(x+x_j)^2 + (y+y_j)^2}]\}, \quad (92)$$

where $\{(x_j, y_j)\} = \{(3, 3), (3, 17), (17, 3), (17, 17)\}$ and $-30 \leq x \leq 10, -30 \leq y \leq 10$.

We note that it is difficult to obtain the analytical solution of sine-Gordon equation, and thus we conducted a comparison of the present numerical results and some available works [18,46]. The physical parameters are specified by $\alpha = 4.0$, $\beta = 4.14$, and $\gamma = 1.0/0.436$, which are the same as those in the previous works [18,46], and additionally, the transformation $w = \partial_t \phi$ is used to translate Eq. (87) into a nonlinear diffusion equation. Here, for simplicity, scheme B with $\mathbf{B} = \mathbf{0}, \mathbf{C} = \mathbf{0}, \mathbf{D} = \phi \mathbf{I}$ is considered, and the equilibrium equation is given by

$$f_i^{\text{eq}} = \omega_i \left[w + \frac{(c_s^2 \phi \mathbf{I} - c_s^2 w \mathbf{I})(\mathbf{c}_i \mathbf{c}_i - c_s^2 \mathbf{I})}{2c_s^4} \right]. \quad (93)$$

Once the macroscopic variable w is determined by $w = \sum_i f_i$, one can obtain the function ϕ with the following explicit difference scheme:

$$\begin{aligned} \phi(x, y, t + \Delta t) &= \Delta t w(x, y, t + \Delta t) + \phi(x, y, t) \\ &= \Delta t \sum_j f_j(x, y, t + \Delta t) + \phi(x, y, t). \end{aligned} \quad (94)$$

We carried out some simulations with $\Delta x = 0.05$, $c = 50$, and $\text{CFL} = 0.1$ and present the results of one circular ring soliton, two circular ring solitons, and four circular ring solitons in Figs. 12–14. From these figures, one can find that the present results are in good agreement with the previous works [18,46].

TABLE III. Example 2: A Comparison of the Computational Time among FDLBM, SRT-FDLBM, and MRT-FDLBM at 1000 steps.

	MRT-FDLBM	SRT-FDLBM	FDLBM
Total time	634.66 s	278.97 s	217.12 s
Ratio	2.92	1.28	1.00

D. Example 4: Two-dimensional heat-conduction equation

As a benchmark problem, the two-dimensional nonlinear heat-conduct equation is also adopted to test the stability of MRT-FDLBM. The mathematical equation of this problem can be expressed as

$$\partial_t \phi - \alpha(\partial_{xx} \phi^\delta + \partial_{yy} \phi^\delta) - \phi + \phi^\delta = 0, \quad \delta > 1. \quad (95)$$

Here we consider two different initial conditions. The first one is

$$\phi(x, y, t) = \left\{ \frac{1}{2} + \frac{1}{2} \tanh \left[\frac{\delta - 1}{2\delta\sqrt{2\alpha}}(x + y) \right] \right\}^{-1/(\delta-1)}, \quad (96)$$

which leads to the following analytical solution:

$$\phi(x, y, t) = \left\{ \frac{1}{2} + \frac{1}{2} \tanh \left[\frac{\delta - 1}{2\delta\sqrt{2\alpha}}(x + y + \sqrt{2\alpha}t) \right] \right\}^{-1/(\delta-1)}. \quad (97)$$

The second one is

$$\phi(x, y, t) = \left\{ \frac{1}{2} - \frac{1}{2} \tanh \left[\frac{\delta - 1}{2\delta\sqrt{2\alpha}}(x + y) \right] \right\}^{-1/(\delta-1)}, \quad (98)$$

and the corresponding analytical solution can be given by

$$\phi(x, y, t) = \left\{ \frac{1}{2} - \frac{1}{2} \tanh \left[\frac{\delta - 1}{2\delta\sqrt{2\alpha}}(x + y - \sqrt{2\alpha}t) \right] \right\}^{-1/(\delta-1)}, \quad (99)$$

where δ is set to be 1.5 and α is the diffusion coefficient. In our simulations, the computational domain is $[0, 1] \times [0, 1]$, $\Delta x = 0.01$, and $\text{CFL} = 0.2$. For this problem, scheme B with $\mathbf{B} = \mathbf{0}, \mathbf{C} = \mathbf{0}$ and $\mathbf{D} = \phi^\delta \mathbf{I}$ is also applied.

This problem is used to test the stability of five different LBMs, including LBM, FDLBM, SRT-FDLBM, SMRT-FDLBM, and MRT-FDLBM. We considered two different

TABLE IV. Example 2: The Computational Time among FDLBM, SRT-FDLBM, and MRT-FDLBM within the similarity GRE.

	FDLBM	SRT-FDLBM	MRT-FDLBM
GRE	9.9945×10^{-5}	9.9962×10^{-5}	9.9960×10^{-5}
CFL	0.1	0.4	0.5
Total time	327.87 s	127.16 s	198.56 s
Ratio	1.00	0.39	0.61

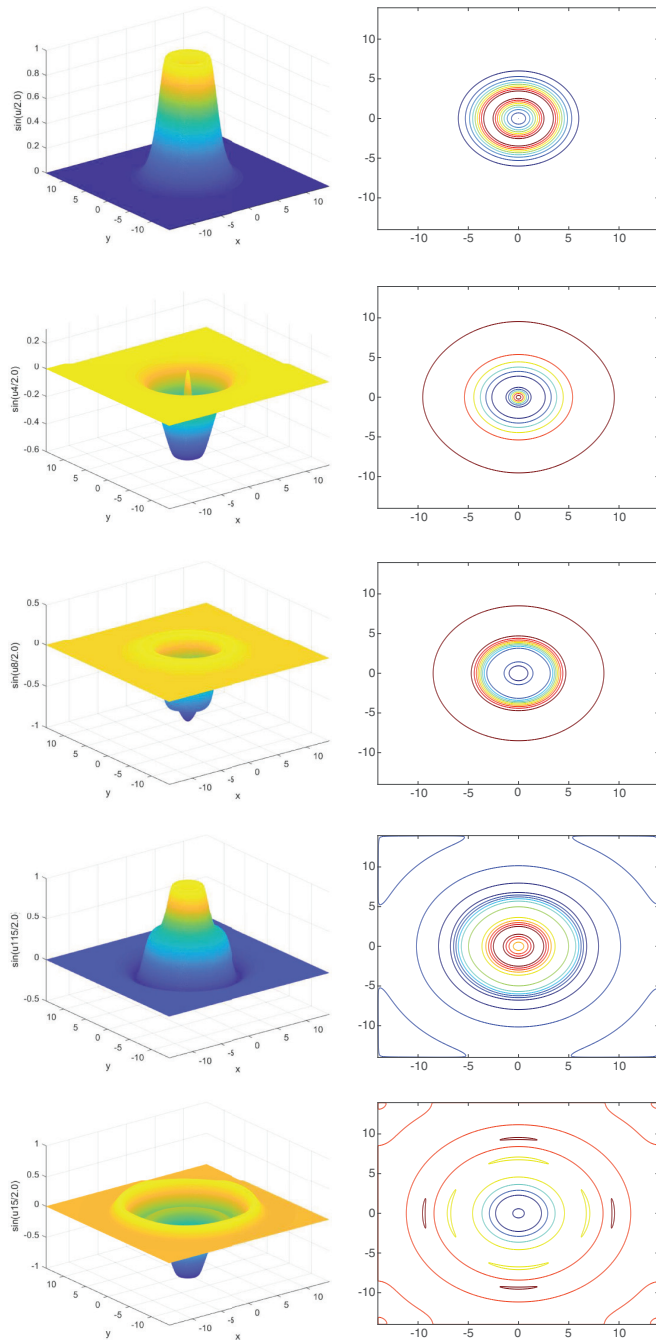


FIG. 12. Example 3: The numerical results of one ring soliton at the time $t = 0, 2, 8, 11.5, 15$ from top to bottom.

initial conditions, i.e., Eqs. (96) and (98) and present the numerical results of different LBMs in Tables VI and VII. As we can from Table VI, for the small diffusion coefficient α and large lattice speed c , LBM is unstable, but FDLBM, SRT-FDLBM, and MRT-FDLBM work well. At the same time, it should be noted that the GREs of SRT-FDLBM and MRT-FDLBM are slightly larger than those of FDLBM at $CFL = 0.2$. In addition, from Table VII, one can find that the FDLBM is unstable for high CFL condition number. However, when $\alpha = 0.08$, SRT-FDLBM with $CFL = 0.5$, SMRT-FDLBM with $CFL = 0.5$, and MRT-FDLBM with $CFL = 0.7$ can work well. When $\alpha = 0.2$, the SRT-FDLBM, SMRT-FDLBM,

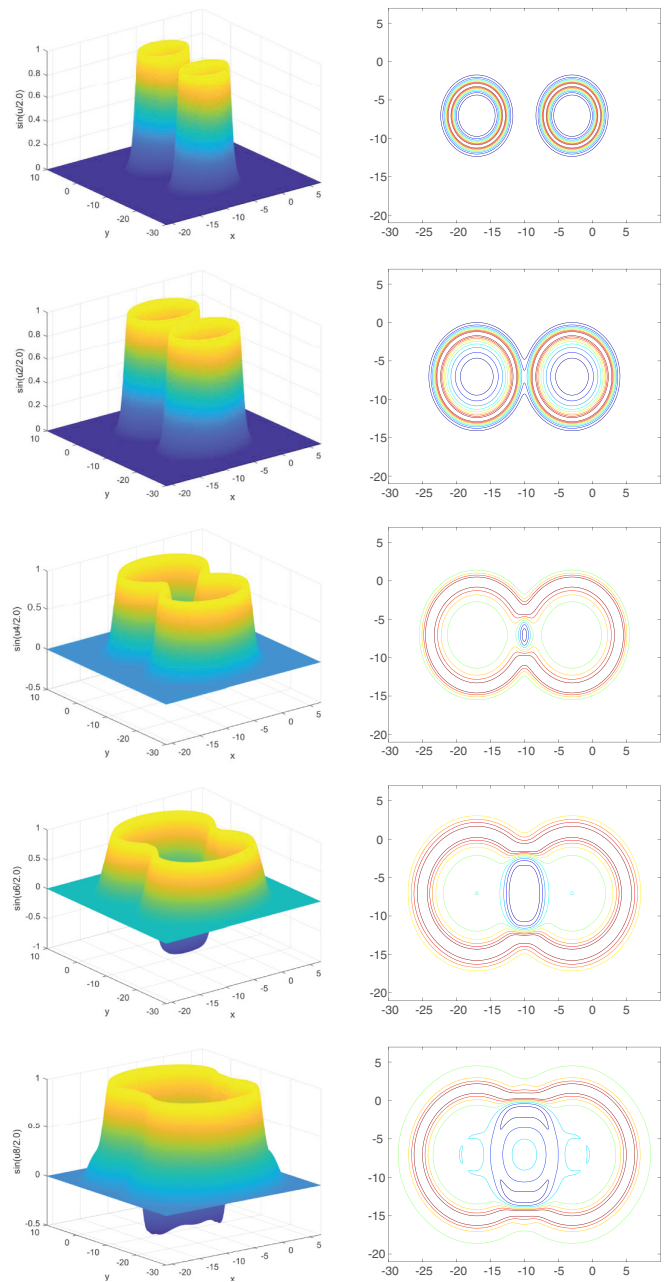


FIG. 13. Example 3: The numerical results of two ring solitons at the time $t = 0, 2, 4, 6, 8$ from top to bottom.

and MRT-FDLBM are still stable for $CFL = 0.3, 0.3,$ and 0.5 . In particular, the stability of the four models is influenced by the diffusion coefficient α and the weight coefficient of the mixed difference scheme η . When α is smaller, the maximum CFL condition numbers of four models are smaller. When α is larger, the central difference scheme will affect the stability of the four models, so the values of CFL condition numbers are still small. Furthermore, it is worth noting that the LBM is unstable when Eq. (98) is adopted as initial condition. The results indicate that for the problem, MRT-FDLBM is the most stable, followed by SMRT-FDLBM, SRT-FDLBM, FDLBM, and standard LBM.

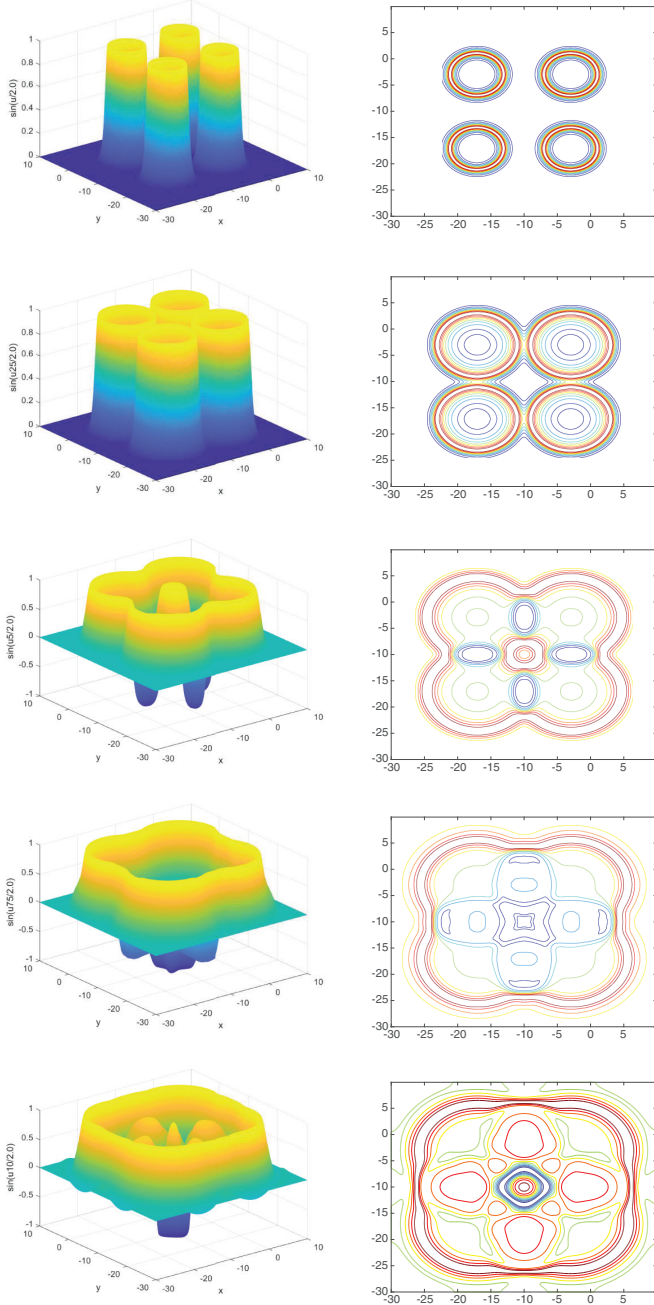


FIG. 14. Example 3: The numerical results of four ring solitons at the time $t = 0, 2.5, 5, 7.5, 10$ from top to bottom.

At last, the computation efficiency of the SMRT-FDLBM, TRT-FDLBM ($\tau_1 = 1.5\tau_2$) and MRT-FDLBM are also tested with the initial condition of Eq. (98). We present the results of three models in Table VIII. From this table, one can observe that TRT-FDLBM and SMRT-FDLBM are more efficient than MRT-FDLBM. The SMRT-FDLBM can save about 15% computational time, and TRT-FDLBM can save about 25%.

E. Example 5: Two-dimensional Schrödinger equation

There are lots of works for solving complex-valued NCDE [18,56,57]. In this part we consider the Schrödinger equation to test the accuracy of MRT-FDLBM. The two-dimensional

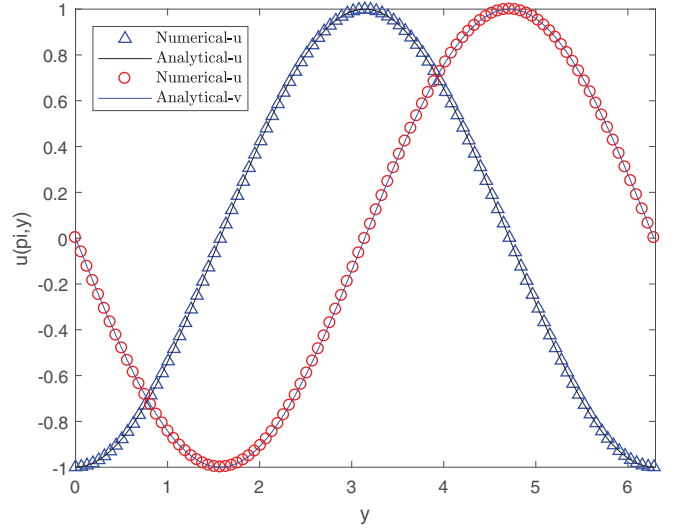


FIG. 15. Example 5: The numerical results of real part (u) and the imaginary part (v) at $t = 1.0$.

nonlinear Schrödinger equation can be written as

$$i\partial_t\phi + \partial_{xx}\phi + \partial_{yy}\phi + \beta|\phi|^2\phi = 0. \quad (100)$$

Under the proper initial and boundary conditions, we can obtain its analytical solution,

$$\phi(x, y, t) = A \exp[i(c_1x + c_2y - nt)], \quad (101)$$

where $n = c_1^2 + c_2^2 - \beta|A|^2$. The physical domain of this problem is $[0, 2\pi] \times [0, 2\pi]$. To solve this problem, we first rewrite Eq. (101) into another form,

$$\partial_t\phi = i(\partial_{xx}\phi + \partial_{yy}\phi) + i\beta|\phi|^2\phi. \quad (102)$$

In our simulations, scheme B with $\mathbf{B} = \mathbf{0}$, $\mathbf{C} = \mathbf{0}$, and $\mathbf{D} = \phi\mathbf{I}$ is adopted; the grid size is 100×100 ; the other parameters are set as $A = c_1 = c_2 = 1$, $\beta = 2$, and $\text{CFL} = 0.1$; and the periodic boundary conditions is used for all boundaries. To test the convergence rates of MRT-FDLBM in space and time, the grid size varies from $2\pi/100$ to $2\pi/140$, and the time step ranges from 0.0002 to 0.0008. From the numerical results in Figs. 15 and 16, one can see that the MRT-FDLBM is not only accurate for solving this complex-valued NCDE but also of second-order convergence rates in space and time. In addition, the Schrödinger equation is also used to test the accuracy of present MRT-FDLBM with LBM. The numerical results are shown in Table IX. As seen from this table, the GREs of MRT-FDLBM are indeed smaller than those of LBM. Thus the present MRT-FDLBM is more accurate than LBM.

VIII. CONCLUSION

In this paper, a second-order MRT-FDLBM is proposed for the general NCDE, and through the CE analysis, the NCDE can be recovered correctly from MRT-FDLBM. The stability of MRT-FDLBM is also analyzed, and it is found the stability region of MRT-FDLBM is larger than that of SRT-FDLBM. However, for MRT-FDLBM, the improvement of stability and accuracy comes at the expense of increased computational

TABLE V. Example 2: The GREs of FDLBM, SRT-FDLBM, SMRT-FDLBM, and MRT-FDLBM with different values of the parameter b and CFL condition number.

	CFL	0.1	0.2	0.3	0.4	0.5
$b = 0.1$	FDLBM	8.7133×10^{-4}	—	—	—	—
	SRT-FDLBM	8.8606×10^{-4}	8.8463×10^{-4}	8.8300×10^{-4}	8.8072×10^{-4}	—
	SMRT-FDLBM	4.1782×10^{-4}	4.2258×10^{-4}	4.2692×10^{-4}	4.3070×10^{-4}	—
	MRT-FDLBM	4.1195×10^{-4}	4.1083×10^{-4}	4.0926×10^{-4}	4.0708×10^{-4}	4.0446×10^{-4}
	CFL	0.1	0.2	0.3	0.4	0.5
$b = 0.15$	FDLBM	—	—	—	—	—
	SRT-FDLBM	—	—	—	—	—
	SMRT-FDLBM	5.6369×10^{-4}	5.6831×10^{-4}	5.7276×10^{-4}	1.9556×10^{-2}	—
	MRT-FDLBM	5.5830×10^{-4}	5.5753×10^{-4}	5.5658×10^{-4}	5.5520×10^{-4}	—

time. Thus, the SMRT-FDLBM is developed to improve the computational efficiency.

Some classic real- and complex-valued NCDEs, including the isotropic convection-diffusion equation, the Burgers-Fisher equation, the sine-Gordon equation, the heat-conduct equation, and the Schrödinger equation, are used to test the performance of MRT-FDLBM, SMRT-FDLBM, and SRT-FDLBM. The numerical results show that MRT-FDLBM is accurate and stable for the real- and complex-valued NCDE, and moreover, both MRT-FDLBM and SMRT-FDLBM have the second-order convergence rates in time and space. Compared to FDLBM and SRT-FDLBM, MRT-FDLBM can be more accurate and stable through adjusting the relaxation parameters properly. In addition, it is also found that the SMRT-FDLBM can improve the computational efficiency with saving about 15% CPU time, and the TRT-FDLBM can save about 25% CPU time. Thus, MRT-FDLBM is more stable while SMRT-FDLBM and TRT-FDLBM are more efficient. If the problem is anisotropic and require stronger stability, then MRT-FDLBM can be a better choice. And if the problem is isotropic and require highly efficient, then both SMRT-FDLBM and TRT-FDLBM are better options. Finally, we point out that the MRT-FDLBM can also be extended to solve the Navier-Stokes equations, which would be considered in a future work.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (Grants No. 51836003

and No. 12072127), the Graduates Innovation Fund, Huazhong University of Science and Technology (Grant No. 2020yjsCXCY034), and the Nature Science Foundation of Hubei province (Grant No. 2020CFB384).

APPENDIX A: THE DISCRETE VELOCITY MODEL AND THE COLLISION MATRIX

In the MRT-FDLBM, there are some different discrete velocity models, here only some popular ones are listed.

D1Q3:

$$\mathbf{c}_j = (0, 1, -1)c, \tag{A1a}$$

$$\omega_0 = 2/3, \omega_1 = \omega_2 = 1/6, \tag{A1b}$$

D2Q9:

$$\mathbf{c}_j = \begin{pmatrix} 0 & 1 & 0 & -1 & 0 & 1 & -1 & -1 & 1 \\ 0 & 0 & 1 & 0 & -1 & 1 & 1 & -1 & -1 \end{pmatrix} c, \tag{A2a}$$

$$\omega_0 = 4/9, \omega_{j=1-4} = 1/9, \omega_{j=5-9} = 1/36, \tag{A2b}$$

D3Q19:

TABLE VI. Example 4: A Comparison of GREs among LBM, FDLBM, and MRT-FDLBM under the Initial Condition (96).

α	Model	$c = 10$	$c = 20$	$c = 100$
$\alpha = 0.005$	LBM	—	—	—
	FDLBM	1.9896×10^{-4}	2.3295×10^{-4}	4.9850×10^{-4}
	SRT-FDLBM	1.1869×10^{-3}	1.8009×10^{-4}	7.5597×10^{-4}
	MRT-FDLBM	2.0664×10^{-3}	3.0980×10^{-3}	3.7663×10^{-3}
$\alpha = 0.01$	LBM	1.8458×10^{-5}	—	—
	FDLBM	1.7215×10^{-4}	1.8401×10^{-4}	3.1222×10^{-4}
	SRT-FDLBM	8.4498×10^{-5}	1.0199×10^{-4}	3.2486×10^{-4}
	MRT-FDLBM	8.4498×10^{-5}	1.0199×10^{-4}	3.2486×10^{-4}
$\alpha = 0.5$	LBM	9.2449×10^{-5}	2.1623×10^{-5}	—
	FDLBM	1.0810×10^{-4}	5.7269×10^{-5}	6.9485×10^{-5}
	SRT-FDLBM	1.8492×10^{-4}	4.7922×10^{-5}	1.5844×10^{-5}
	MRT-FDLBM	1.6859×10^{-4}	3.6927×10^{-5}	6.8443×10^{-5}

TABLE VII. Example 4: The GREs of LBM, FDLBM, SRT-FDLBM, and MRT-FDLBM under the Initial Condition (98) ($t = 1.0$).

α	Model	CFL = 0.1	CFL = 0.3	CFL = 0.5	CFL = 0.7	CFL = 0.9
$\alpha = 0.003$	FDLBM	1.5466×10^{-2}	—	—	—	—
	SRT-FDLBM	1.5303×10^{-2}	1.5388×10^{-2}	—	—	—
	SMRT-FDLBM	1.5324×10^{-2}	1.5393×10^{-2}	—	—	—
	MRT-FDLBM	5.1796×10^{-2}	8.6652×10^{-2}	—	—	—
$\alpha = 0.08$	FDLBM	4.7437×10^{-4}	—	—	—	—
	SRT-FDLBM	4.7196×10^{-4}	4.7148×10^{-4}	4.6952×10^{-4}	—	—
	SMRT-FDLBM	4.5654×10^{-4}	4.5474×10^{-4}	4.5111×10^{-4}	—	—
	MRT-FDLBM	9.1571×10^{-4}	1.9478×10^{-3}	2.9333×10^{-3}	7.8531×10^{-3}	—
$\alpha = 0.2$	FDLBM	—	—	—	—	—
	SRT-FDLBM	4.2587×10^{-4}	4.2707×10^{-4}	—	—	—
	SMRT-FDLBM	3.0619×10^{-4}	3.0903×10^{-4}	—	—	—
	MRT-FDLBM	2.5429×10^{-4}	2.2507×10^{-4}	3.8161×10^{-4}	—	—

$$\mathbf{c}_j = \begin{pmatrix} 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 \end{pmatrix} \mathbf{c}, \quad (\text{A3a})$$

$$\omega_0 = 1/3, \omega_{j=1-6} = 1/18, \omega_{j=7-18} = 1/36. \quad (\text{A3b})$$

The collision matrix $\tilde{\Lambda}$ is given by

$$\tilde{\Lambda} = \mathbf{M}^{-1} \mathbf{S} \mathbf{M}, \quad (\text{A4a})$$

$$\mathbf{S} = (s_{kj}), s_{kj} = 0 \quad (k \neq j), s_{kk} = s_k. \quad (\text{A4b})$$

In this work, we only take the D2Q9 lattice model as an example, and the transportation matrix \mathbf{M} can be expressed as

$$\mathbf{M} = \mathbf{C}_d \mathbf{M}_0, \quad (\text{A5})$$

where $\mathbf{C}_d = \text{diag}(c^0, c^2, c^4, c^1, c^3, c^1, c^3, c^2, c^2)$ and $\mathbf{S} = \text{diag}(s_0, s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8)$ are diagonal matrixs, \mathbf{M}_0 is given as [21]

$$\mathbf{M}_0 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -4 & -1 & -1 & -1 & -1 & 2 & 2 & 2 & 2 \\ 4 & -2 & -2 & -2 & -2 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & -1 & 0 & 1 & -1 & -1 & 1 \\ 0 & -2 & 0 & 2 & 0 & 1 & -1 & -1 & 1 \\ 0 & 0 & 1 & 0 & -1 & 1 & 1 & -1 & -1 \\ 0 & 0 & -2 & 0 & 2 & 1 & 1 & -1 & -1 \\ 0 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \end{pmatrix}. \quad (\text{A6})$$

In addition, the diagonal matrix $\tilde{\mathbf{S}}$ related to the diffusion matrix \mathbf{K} can be written as

$$\tilde{\mathbf{S}} = \begin{pmatrix} s_3 & 0 \\ 0 & s_5 \end{pmatrix}. \quad (\text{A7})$$

It should be noted that for the complex-valued NCDE, $\kappa = \kappa_r + i\kappa_c$, $\tau = \tau_r + i\tau_c$, and the following requirements should be satisfied:

$$\tau_r = \frac{\kappa_r}{c_s^2}, \quad \tau_c = \frac{\kappa_c}{c_s^2}, \quad (\text{A8})$$

$$\tilde{\Lambda}_r = \mathbf{M}^{-1} \mathbf{S}_r \mathbf{M}, \quad \tilde{\Lambda}_c = \mathbf{M}^{-1} \mathbf{S}_c \mathbf{M}, \quad (\text{A9})$$

$$\mathbf{S}_r = \text{diag}(s_{r0}, s_{r1}, s_{r2}, s_{r3}, s_{r4}, s_{r5}, s_{r6}, s_{r7}, s_{r8}),$$

$$\mathbf{S}_c = \text{diag}(s_{c0}, s_{c1}, s_{c2}, s_{c3}, s_{c4}, s_{c5}, s_{c6}, s_{c7}, s_{c8}),$$

where the relaxation parameters are given by

$$s_{r3} = s_{r5} = \frac{\tau_r}{\tau_r^2 + \tau_c^2}, \quad s_{c3} = s_{c5} = \frac{-\tau_c}{\tau_r^2 + \tau_c^2}, \quad (\text{A10})$$

TABLE VIII. Example 4: A Comparison of the GREs and CPU Time between MRT-FDLBM and SMRT-FDLBM.

		$\alpha = 0.1$		$\alpha = 0.5$	
		$c = 10$	$c = 20$	$c = 10$	$c = 20$
MRT-FDLBM	GRE	4.3708×10^{-4}	1.1080×10^{-3}	5.3055×10^{-4}	7.9273×10^{-5}
	CPU time	40.42	80.11	39.09	77.65
SMRT-FDLBM	GRE	4.4798×10^{-4}	4.2090×10^{-4}	5.4335×10^{-4}	8.9687×10^{-5}
	CPU time	33.76	67.49	32.81	64.49
Decrease percentage		16.48%	15.75%	16.07%	16.95%
TRT-FDLBM	GRE	4.3580×10^{-3}	1.4924×10^{-2}	6.8148×10^{-4}	4.0286×10^{-4}
	CPU time	29.32	58.55	29.22	58.37
Decrease percentage		27.46%	26.91%	25.25%	24.83%

APPENDIX B: THE CALCULATION PROCESS OF THE TRT-FDLBM

The evolution equation of TRT-FDLBM is the same as Eqs. (10) and (16), but some of the distribution functions are defined differently from MRT-FDLBM, including $\hat{f}_j, \hat{f}_j^+, \bar{f}_j, \bar{f}_j^+$. They can be defined as follows:

$$\hat{f}_j = f_j - \frac{1}{2}\Delta t [-w_1(f_j - f_j^{eq}) - w_2(\bar{f}_j - \bar{f}_j^{eq})], \tag{B1a}$$

$$\hat{f}_j^+ = f_j + \frac{1}{2}\Delta t [-w_1(f_j - f_j^{eq}) - w_2(\bar{f}_j - \bar{f}_j^{eq})], \tag{B1b}$$

$$\bar{f}_j = f_j - \frac{1}{2}h [-w_1(f_j - f_j^{eq}) - w_2(\bar{f}_j - \bar{f}_j^{eq})], \tag{B2a}$$

$$\bar{f}_j^+ = f_j + \frac{1}{2}h [-w_1(f_j - f_j^{eq}) - w_2(\bar{f}_j - \bar{f}_j^{eq})], \tag{B2b}$$

where $w_1 = (\frac{1}{\tau_1} + \frac{1}{\tau_2})/2$ and $w_2 = (\frac{1}{\tau_1} - \frac{1}{\tau_2})/2$, \bar{j} represents the opposite direction from j . For NCDE, τ_1 is an adjustable relaxation parameter, τ_2 is related to the diffusion coefficient and satisfy $\tau_2 c_s^2 = K$. In addition, the key calculations need to be adjusted as follows:

$$\hat{f}_j^+ = P_{11}\hat{f}_j + P_{12}\hat{f}_{\bar{j}} + P_{13}f_j^{eq} + P_{14}f_{\bar{j}}^{eq}, \tag{B3}$$

where

$$P_{11} = \frac{4 + \Delta t^2(-w_1^2 + w_2^2)}{4 + 4\Delta t w_1 + \Delta t^2(w_1^2 - w_2^2)}, \quad P_{12} = \frac{-4\Delta t}{4 + 4\Delta t w_1 + \Delta t^2(w_1^2 - w_2^2)}, \tag{B4}$$

$$P_{13} = \frac{2\Delta t(2w_1 + \Delta t w_1^2 - \Delta t w_2^2)}{4 + 4\Delta t w_1 + \Delta t^2(w_1^2 - w_2^2)}, \quad P_{14} = -P_{12}. \tag{B5}$$

$$\bar{f}_j^+ = P_{21}\hat{f}_j + P_{22}\hat{f}_{\bar{j}} + P_{23}f_j^{eq} + P_{24}f_{\bar{j}}^{eq}, \tag{B6}$$

where

$$P_{21} = \frac{8 + 2\Delta t w_1 + \Delta t^2(-w_1^2 + w_2^2)}{8 + 8\Delta t w_1 + 2\Delta t^2(w_1^2 - w_2^2)}, \quad P_{22} = \frac{-6\Delta t}{8 + 8\Delta t w_1 + 2\Delta t^2(w_1^2 - w_2^2)}, \tag{B7}$$

TABLE IX. Example 5: A Comparison of the GREs between MRT-FDLBM and LBM.

		$A = 0.5$	$A = 1.0$	$A = 1.5$	$A = 2.0$
$\beta = 2.0$	LBM	4.8457×10^{-2}	8.4781×10^{-2}	5.8348×10^{-1}	—
	MRT-FDLBM	1.0915×10^{-3}	2.1201×10^{-3}	7.3488×10^{-3}	1.3168×10^{-2}
	LBM	4.7721×10^{-2}	6.2417×10^{-2}	1.8778×10^{-1}	—
$\beta = 1.5$	MRT-FDLBM	1.1981×10^{-3}	1.3396×10^{-3}	4.9498×10^{-3}	9.9948×10^{-3}
	LBM	4.7266×10^{-2}	5.3879×10^{-2}	1.0455×10^{-1}	3.8765×10^{-1}
$\beta = 1.0$	MRT-FDLBM	1.3185×10^{-3}	9.2493×10^{-4}	2.5817×10^{-3}	6.3101×10^{-3}
	LBM	4.6952×10^{-2}	4.8457×10^{-2}	5.5614×10^{-2}	8.4781×10^{-2}
$\beta = 0.5$	MRT-FDLBM	1.4490×10^{-3}	1.0915×10^{-3}	9.7717×10^{-4}	2.1201×10^{-3}

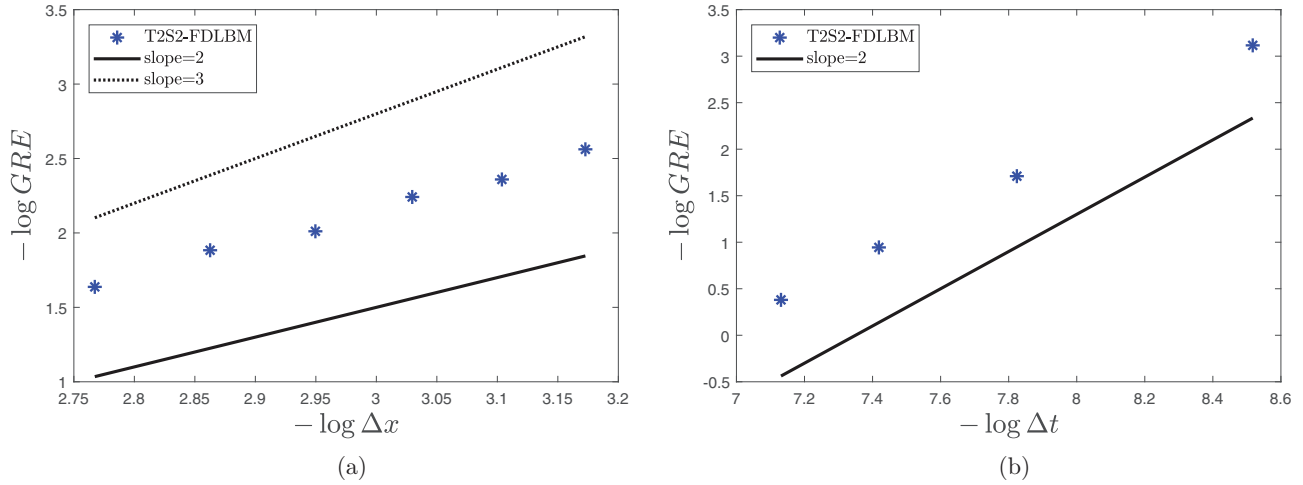


FIG. 16. Example 5: GREs of MRT-FDLBM. (a) GREs at different grid sizes and (b) GREs with different time steps.

$$P_{23} = \frac{3\Delta t(2w_1 + \Delta t w_1^2 - \Delta t w_2^2)}{8 + 8\Delta t w_1 + 2\Delta t^2(w_1^2 - w_2^2)}, \quad P_{24} = -P_{22}. \quad (\text{B8})$$

$$f_j = P_{31}\tilde{f}_j + P_{32}\tilde{f}_j + P_{33}f_j^{\text{eq}} + P_{34}f_j^{\text{eq}}, \quad (\text{B9})$$

where

$$P_{31} = \frac{16 + 4\Delta t w_1}{16 + 8\Delta t w_1 + \Delta t^2(w_1^2 - w_2^2)}, \quad P_{32} = \frac{-4\Delta t w_2}{4 + 4\Delta t w_1 + \Delta t^2(w_1^2 - w_2^2)}, \quad (\text{B10})$$

$$P_{33} = \frac{\Delta t(4w_1 + \Delta t w_1^2 - \Delta t w_2^2)}{16 + 8\Delta t w_1 + \Delta t^2(w_1^2 - w_2^2)}, \quad P_{34} = -P_{32}. \quad (\text{B11})$$

-
- [1] E. L. Cussler, *Diffusion: Mass Transfer in Fluid Systems* (Cambridge University Press, Cambridge, UK, 2009).
- [2] R. J. LeVeque, *Finite Volume Methods for Hyperbolic Problems*, Vol. 31 (Cambridge University Press, Cambridge, UK, 2002).
- [3] C. Johnson, *Numerical Solution of Partial Differential Equations by the Finite Element Method* (Courier Corporation, North Chelmsford, MA, 2009).
- [4] J. W. Thomas, *Numerical Partial Differential Equations: Finite Difference Methods*, Vol. 22 (Springer Science and Business Media, New York, 2013).
- [5] S. Succi, E. Foti, and F. Higuera, *Europhys. Lett.* **10**, 433 (1989).
- [6] A. Montessori, P. Prestininzi, M. La Rocca, and S. Succi, *Phys. Rev. E* **92**, 043308 (2015).
- [7] X. Shan and H. Chen, *Phys. Rev. E* **47**, 1815 (1993).
- [8] M. R. Swift, W. R. Osborn, and J. M. Yeomans, *Phys. Rev. Lett.* **75**, 830 (1995).
- [9] G. Falcucci, S. Ubertini, and S. Succi, *Soft Matter* **6**, 4357 (2010).
- [10] Y. Gan, A. Xu, G. Zhang, and S. Succi, *Soft Matter* **11**, 5336 (2015).
- [11] H. Chen, S. Kandasamy, S. Orszag, R. Shock, S. Succi, and V. Yakhot, *Science* **301**, 633 (2003).
- [12] G. Strumolo and V. Babu, *Phys. World* **10**, 45 (1997).
- [13] A. Xu, G. Zhang, and Y. Ying, *Acta Phys. Sin.* **64**, 184701 (2015).
- [14] C. Lin, A. Xu, G. Zhang, and Y. Lin, *Combust. Flame* **164**, 137 (2016).
- [15] C. Lin, A. Xu, G. Zhang, K. H. Luo, and Y. Li, *Phys. Rev. E* **96**, 053305 (2017).
- [16] Y. Gan, A. Xu, G. Zhang, C. Lin, H. Lai, and Z. Liu, *Front. Phys.* **14**, 43602 (2019).
- [17] S. P. Dawson, S. Chen, and G. D. Doolen, *J. Chem. Phys.* **98**, 1514 (1993).
- [18] B. Shi and Z. Guo, *Phys. Rev. E* **79**, 016701 (2009).
- [19] B. Chopard, J. L. Falcone, and J. Latt, *Eur. Phys. J. Spec. Top.* **171**, 245 (2009).
- [20] H. Yoshida and M. Nagaoka, *J. Comput. Phys.* **229**, 7774 (2010).
- [21] Z. Chai, B. Shi, and Z. Guo, *J. Sci. Comput.* **69**, 355 (2016).
- [22] I. Ginzburg, *Adv. Water Resour.* **28**, 1196 (2005).
- [23] I. Ginzburg, *J. Stat. Phys.* **126**, 157 (2007).
- [24] I. Ginzburg, *Commun. Comput. Phys.* **11**, 1439 (2012).
- [25] I. Ginzburg, *Adv. Water Resour.* **51**, 381 (2013).
- [26] Y. Zhao, Y. Wu, Z. Chai, and B. Shi, *Comput. Math. Appl.* **79**, 2550 (2020).
- [27] Z. Chai and B. Shi, *Phys. Rev. E* **102**, 023306 (2020).
- [28] X. He and G. Doolen, *J. Comput. Phys.* **134**, 306 (1997).
- [29] X. He and G. D. Doolen, *Phys. Rev. E* **56**, 434 (1997).

- [30] R. van der Sman and M. Ernst, *J. Comput. Phys.* **160**, 766 (2000).
- [31] X. Zhang, A. Bengough, J. Crawford, and I. Young, *Adv. Water Resour.* **25**, 1 (2002).
- [32] J. Lu, Z. Chai, B. Shi, Z. Guo, and G. Hou, *Phil. Trans. R. Soc. A* **369**, 2311 (2011).
- [33] M. B. Reider and J. D. Sterling, *Comput. Fluids* **24**, 459 (1995).
- [34] R. Mei and W. Shyy, *J. Comput. Phys.* **143**, 426 (1998).
- [35] M. Junk and A. Klar, *SIAM J. Sci. Comput.* **22**, 1 (2000).
- [36] Z. Guo and T. S. Zhao, *Phys. Rev. E* **67**, 066709 (2003).
- [37] Y. Wang, Y. He, J. Huang, and Q. Li, *Int. J. Numer. Methods Fluids* **59**, 853 (2009).
- [38] K. Hejranfar and E. Ezzatneshan, *Int. J. Numer. Methods Fluids* **75**, 713 (2014).
- [39] K. Hejranfar and M. H. Saadat, *Comput. Math. Appl.* **76**, 1427 (2018).
- [40] E. Ezzatneshan and K. Hejranfar, *Int. J. Numer. Methods Fluids* **89**, 235 (2019).
- [41] K. Hejranfar and E. Ezzatneshan, *Phys. Rev. E* **92**, 053305 (2015).
- [42] S. Polasanapalli and K. Anupindi, *Comput. Fluids* **181**, 259 (2019).
- [43] H. R. Khakrah, P. Hooshmand, D. Ross, and M. Jamshidian, *Int. J. Numer. Methods Heat Fluid Flow* **30**, 977 (2019).
- [44] M. Sakthivel and K. Anupindi, *Phys. Rev. E* **100**, 043307 (2019).
- [45] X. Chen, Z. Chai, H. Wang, and B. Shi, *Comput. Math. Appl.* **80**, 3066 (2020).
- [46] H. Wang, B. Shi, H. Liang, and Z. Chai, *Appl. Math. Comput.* **309**, 334 (2017).
- [47] S. A. Hosseini, N. Darabiha, D. Thévenin, and A. Eshghinejadfard, *Int. J. Mod. Phys. C* **28**, 1750141 (2017).
- [48] P. Lallemand and L.-S. Luo, *Phys. Rev. E* **61**, 6546 (2000).
- [49] C. Coreixas, G. Wissocq, B. Chopard, and J. Latt, *Phil. Trans. R. Soc. A* **378**, 20190397 (2020).
- [50] J. Sterling and S. Chen, *J. Comput. Phys.* **123**, 196 (1996).
- [51] S. A. Hosseini, C. Coreixas, N. Darabiha, and D. Thevenin, *Phys. Rev. E* **99**, 063305 (2019).
- [52] A. D. Rosis and C. Coreixas, *Phys. Fluids* **32**, 117101 (2020).
- [53] I. Ginzburg, D. d’Humières, and A. Kuzmin, *J. Stat. Phys.* **139**, 1090 (2010).
- [54] Z. Guo, C. Zheng, and B. Shi, *Chin. Phys.* **11**, 366 (2002).
- [55] H. L. Wang, Z. H. Chai, B. C. Shi, and H. Liang, *Phys. Rev. E* **94**, 033304 (2016).
- [56] S. Succi, *Phys. Rev. E* **53**, 1969 (1996).
- [57] P. J. Dellar, D. Lapitski, S. Palpacelli, and S. Succi, *Phys. Rev. E* **83**, 046706 (2011).